

Байкова И.В., Копытов М.А., Кулагин М.В.,
Михайлов Г.М., Привезенцев Ю.А., Рогов Ю.П.

Распределенные информационно-вычислительные системы

Выпуск 1

Локальная сеть ВЦ РАН

Оглавление

| | |
|--|----|
| Введение | 2 |
| 1. Архитектура локальной сети. | 3 |
| 1.1 Концепция выбора архитектуры сети..... | 3 |
| 1.2 Топология сетей. | 4 |
| 1.2.1 Топология шины. | 4 |
| 1.2.2 Топология звезды. | 4 |
| 1.2.3 Топология кольца..... | 5 |
| 1.3 Протоколы доступа. Международные стандарты. | 6 |
| 1.3.1 Метод множественного доступа с контролем несущей и обнаружения столкновений (CSMA/CD(802.3)). | 6 |
| 1.3.2 Кольцо с передачей маркера (802.5) | 6 |
| 1.3.3 Шина с маркером (802.4)..... | 7 |
| 1.3.4 Сравнительный анализ стандартов..... | 7 |
| 1.3.5 Выбор топологии и стандартов..... | 8 |
| 1.4 Сетевой и транспортный уровень модели OSI. | 9 |
| 1.4.1 TCP/IP | 9 |
| 1.4.2 Производительность транспортного протокола..... | 10 |
| 1.5 Центр коммутации. | 11 |
| 1.6 Физическая среда передачи на витых парах. Исследования и реализация..... | 14 |
| 2. Рабочие станции фирмы SUN Microsystems..... | 16 |
| 2.1 Аппаратные особенности рабочих станций фирмы SUN..... | 16 |
| 2.2 Программная среда рабочих станций и ЭВМ фирмы SUN. | 18 |
| 3. Вычислительная система PARSYTEC GCel 1/64. | 20 |
| 3.1 Аппаратные особенности архитектуры GCel 1/64. | 20 |
| 3.2 Системное программное обеспечение GCel PARIX. | 22 |
| 4. Принципы построения ИВС..... | 25 |
| 4.1 Особенности эволюции ИВС в 70 - 80 годах (обзор). | 25 |
| 4.2 Функциональная структура программно-аппаратной среды современных информационно-вычислительных систем. | 31 |
| 4.2.1 Функциональные области APP. | 32 |
| 4.2.2 Область функций операционной системы. | 32 |
| 4.2.3 Человеко-машинные интерфейсы. | 32 |
| 4.2.4 Функциональная область поддержки разработки программного обеспечения..... | 33 |
| 4.2.5 Функциональная область управления данными..... | 33 |
| 4.2.6 Функциональная область обмена данными. | 34 |
| 4.2.7 Область графических функций. | 34 |
| 4.2.8 Функциональная область сетевой поддержки..... | 35 |
| 4.2.9 Интегрально поддерживаемые функциональные области. | 35 |
| 5. Поддержка, сопровождение и администрирование в ИВС ВЦ РАН..... | 35 |
| 5.1. Общие вопросы. | 35 |
| 5.2. Организация работы с внешней памятью в среде ИВС ВЦ РАН. | 37 |
| 5.2.1. Аппаратные средства внешней памяти и их настройка..... | 37 |
| 5.2.2. Работа с гибкими и компакт-дисками. | 40 |
| 5.2.3. Дисковый сервер, бездисковые и другие клиенты локальной сети..... | 41 |
| 5.2.4. Доступ к файловой системе с персонального компьютера. | 41 |
| 5.3. Пользователи сети, распределение ресурсов и их учет. | 42 |
| 5.3.1. Общие положения. | 42 |

| | |
|--|----|
| 5.3.2. Регистрация пользователей и выделение дисковых квот..... | 42 |
| 5.3.3. Сетевая Информационная Система (NIS)..... | 43 |
| 5.3.4. Учет использования ресурсов на рабочей станции сети. | 44 |
| 5.4. Последовательность работ системного администратора при развертывании сети и ее поддержке..... | 45 |
| Заключение..... | 46 |

Введение

Развитие вычислительной техники в последние десятилетия характеризуется резким ростом производительности, информационной емкости и гибкости информационно-вычислительных систем (ИВС) при беспрецедентном снижении удельной стоимости аппаратных средств. Характерно также и чрезвычайное увеличение парка индивидуальных систем, производительность и информационная емкость которых сегодня заметно выше, чем у больших машин предыдущего поколения. За прошедшее десятилетие накоплены колоссальные информационные ресурсы, включающие гигантские объемы программного обеспечения и всевозможные базы данных, содержащие информацию практически по всем отраслям человеческой деятельности. При резко возросшей производительности ИВС все более существенную роль в процессе разработки программного обеспечения играют различные инструментальные системы и системы автоматизации программирования, языки высокого и сверхвысокого уровня (4GL, CASE-системы и т.п).

Следует подчеркнуть, что чрезвычайно быстрое улучшение характеристик аппаратуры при одновременном снижении ее стоимости приводит к драматически быстрой смене аппаратных платформ при значительно более высокой стабильности как прикладных, так и системных программных средств, стоимость и объем которых постоянно увеличиваются.

Для современных информационных технологий характерно еще и то, что ИВС из средства, доступного ранее сравнительно узкому кругу специалистов, стали повседневным инструментом для рядовых потребителей вне зависимости от рода их деятельности. В течение последних лет для них создано множество прикладных систем, рассчитанных на пользователей, не имеющих профессиональной подготовки в информатике. Решающую роль в этом процессе сыграло создание и массовое внедрение персональных ЭВМ.

Одной из характернейших черт современных ИВС является их интегрированность в обширные и разветвленные информационные структуры, которые уже сегодня имеют глобальные масштабы. Формируется информационная среда, которая уже объединяет сотни тысяч различных вычислительных систем во множестве организаций, решающих свои собственные задачи. Взаимодействие отдельных систем и их компонент в такой среде невозможно без соблюдения единых требований, определяющих такую среду, как единое информационное пространство.

На протяжении последних лет сформировались определенные концепции, на которых сегодня строится процесс применения и развития информационных технологий и которые учитываются всеми участниками процесса информатизации общества: пользователями, поставщиками и разработчиками аппаратуры и программных продуктов.

Реализация концепций построения ИВС существенно зависит от того, на какой базовой основе будет построена такая система.

В рамках проекта "Локальная сеть ВЦ РАН. Развертывание, развитие, программно-аппаратная поддержка" рассмотрены основные этапы проектирования базового уровня ИВС – локальной сети Вычислительного Центра РАН (ЛВС ВЦ РАН). Проектирование базового уровня – это выбор физической среды, протоколов сети всех уровней, реальных структур системы и ее компонентов, программно-аппаратного обеспечения системы в целом.

Построение базового уровня системы безусловно должно быть увязано с запросами создаваемой ИВС и ориентировано как на освоение новых перспективных информационных технологий, так и на те научные проблемы, для решения которых она предназначена.

Среди множества проблем при проектировании базового уровня ЛВС ВЦ РАН основополагающими являются следующие:

- создание коммуникационной среды;
- выбор архитектуры сети;
- выбор основных компонентов сети (файловых, терминальных и вычислительных серверов последовательного, параллельного и конвейерного типов, средств управления локальной сетью с выходом в глобальную сеть, сетевого менеджмента и т.д.);
- выбор и развертывание необходимого набора традиционных систем программирования;
- выбор и развертывание современных баз данных.

Каждая из перечисленных проблем включает в себя не только решения в плане идеологии, но и требует вполне определенных капиталовложений. Поэтому в условиях стремительного развития компьютерного рынка исключительно важное значение приобретает стратегически правильный выбор базового уровня и концепций его развития.

Ниже излагаются подходы авторов при выборе базового уровня ЛВС ВЦ РАН, полученные результаты и принципы организации ИВС на реализованных базовых средствах сети. При этом в результате проведенных исследований таких систем авторы пришли к следующим основным выводам:

- основным средством доступа пользователя к распределенной вычислительной системе должен быть персональный компьютер;
- серверная система первой очереди проекта базируется на рабочих станциях фирмы SUN Microsystems (США);
- в качестве вычислительных серверов помимо рабочих станций должны использоваться высокопроизводительные системы, в том числе параллельной архитектуры, построенные на транспьютерной технологии. В дальнейшем должно быть предусмотрено наращивание спектра вычислительных серверов мощными вычислителями конвейерного типа, не изменяя при этом общей концепции системы;
- коммутационная среда должна быть построена на базе структурированной кабельной сети с использованием интеллектуальных средств коммутации современных технологий с соответствующими протоколами.
- базовое программное обеспечение должно быть только лицензионным и обеспечивать пользователям широкий набор прикладных языков программирования и других программных интерфейсов и средств поддержки для обслуживающего персонала.

1. Архитектура локальной сети.

1.1 Концепция выбора архитектуры сети.

Как было изложено выше, в основу выбора концепции положены, прежде всего, международные стандарты, которые действуют в области информационных технологий и открытых систем в настоящее время. Изначально локальная вычислительная сеть ВЦ РАН (ЛВС ВЦ РАН) проектировалась, исходя из статуса института в масштабе СССР и международном плане. Она должна была быть построена на следующих принципах:

- проектируемая сеть должна быть гетерогенной и должна удовлетворять требованиям открытых систем;
- все компоненты сети должны соответствовать международным стандартам на физическом и протокольном уровнях;

- имеющийся ко времени реализации проекта парк вычислительных средств, в первую очередь, спектр PC 286,386,486, рабочих станций SUN и др., должен быть охвачен сетью с точки зрения построения среды передачи данных;
- все компоненты сети, включая серверы, рабочие станции, абонентские персональные компьютеры (ПК) с адаптерами, программное обеспечение как сетевое, так и общесистемное, должны быть лицензированными;
- в концептуальном плане в основу должен быть положен принцип "клиент-сервер". Типы серверов, их количественные и качественные характеристики должны быть адекватны к требованиям пользователей всех уровней;
- сеть должна иметь в качестве хотя бы одного из серверов мощный вычислитель параллельной или конвейерной архитектуры, исходя из требований для решения задач математической физики, которые традиционно для ВЦ РАН были актуальны и требуют значительную часть всех вычислительных ресурсов.

Прежде чем выбрать архитектуру локальной сети, рассмотрим несколько общих для всех сетей вопросов, без решения которых не представляется возможным данный выбор.

1.2 Топология сетей.

В выборе архитектуры сети одним из актуальных вопросов является выбор топологии сети. Ограничимся кратким изложением наиболее распространенных структур. Существуют четыре базовые топологии:

- топология шины;
- топология звезды;
- топология шины и с ветвями типа звезды;
- топология кольца.

1.2.1 Топология шины.

Одной из простейших для установки проводных схем является топология шины. Компьютеры, подсоединяемые к сети на основе шины, могут просто подключаться к сетевому кабелю, который проходит от одного конца сети до другого. К недостаткам относится необходимость соблюдать ряд электрических требований. Например, существует ограничение на протяженность кабеля от шины до каждого компьютера. Кроме того, основная шина должна сохраняться нетронутой: разрыв при добавлении отводящего кабеля или при удлинении провода приводит к тому, что вся сеть временно не работает. Хорошо известным примером стандарта сети на базе шины является коаксиальная Ethernet.

1.2.2 Топология звезды.

В этой топологии каждый компьютер соединен через кабель (луч) с центральным узлом, через который проходят все данные. Центральный узел в звездной сети может обладать различной степенью интеллектуальности. Разумный узел может направлять данные непосредственно той рабочей станции, которой они адресованы. Существуют также пассивные узлы, в которых не происходит переключений; сигнал от одного луча передается всем другим лучам звезды. Получающий компьютер должен отбирать свои сообщения из данных, которые получают все компьютеры. Электрически данная топология близка к топологии шины, даже несмотря на то,

что провода соединены в виде звезды. Ее часто называют звезднопроводной шиной.

Стратегическое преимущество звездной топологии заключается в том, что большинство сетевых услуг, локализация неисправностей и изменение проводки имеют место в центральном узле. В центре, где каждый рабочий участок подсоединен кабелем к центральному узлу разводки – концентратору, переконфигурация сети осуществляется простым перемещением кабелей в центральном узле.

Недостаток звездной топологии в том, что необходимо иметь отдельный кабель к каждой машине, которой требуется доступ к сети. Если выходит из строя центральный узел, то перестают работать и все станции, подключенные к нему.

1.2.3 Топология кольца.

Третьей базовой топологией является кольцо. Каждый компьютер соединен как с предшествующими, так и с последующими компьютерами по замкнутому кольцу. В сети на базе кольца данные проходят по кольцу через каждый компьютер, пока либо не достигнут компьютера-получателя, который их перехватит, либо не вернуться к отправителю, который их остановит.

Известным примером кольцевой топологии является маркерное кольцо фирмы IBM. В то время как электрически это действительно кольцо, физически ее проводка представляет собой звезду. Это означает, что кольцо возвращается к центральному узлу между всеми рабочими станциями. От каждого компьютера к этому узлу положен кабель. Фирма IBM называет этот узел как Multistation Access Unit или MAU. Такой тип конфигурации часто называют звездно-проводным кольцом.

Правильный выбор топологии с учетом реализации среды передачи данных на физическом уровне является исключительно важным не только по концептуальным соображениям, но и в экономическом плане. Мировая практика показывает, что стоимость капиталовложений по прокладыванию и инсталляции сети достигает 50% от общей стоимости проекта развертывания локальной сети в учреждениях. В условиях ВЦ РАН вопрос выбора топологии сети тоже оказался не очевидным, т.к. к моменту начала реализации проекта локальной сети в здании института были развернуты две параллельные сети:

- сеть на классических неэкранированных витых парах (Unshielded Twisted Pair, UTP) с частичной инсталляцией;
- сеть на коаксиальном кабеле типа РК-75-1.5/11 с волновым сопротивлением 75 ом без инсталляции.

Более подробное описание этих сетей дано в части 1.6. Здесь же отметим, что если сеть UTP проектировалась на скорость 10 Мб/сек при ограничении длины лучей звездной топологии до 150 м, то сеть на коаксиальном кабеле по частотным и электрическим характеристикам более близка к широкополосному кабелю IBM-75, чем к известному стандарту RG-58 (Thin Ethernet) и имеет расчетную скорость передачи данных до 100 Мб/сек.

Учитывая перспективы развития локальной сети, прежде всего, с точки зрения включения в нее мощных высокопроизводительных серверов, а также выхода проектируемой сети к магистральным каналам связи (backbone) со скоростью до 100 Мб/сек и более, в архитектуре сети должно быть предусмотрено место оптоволоконным кабелям связи с распределенным интерфейсом данных (Fibre Distributed Data Interface – FDDI).

Однако в реальных сетях различные участки могут быть выполнены по различным топологиям на различных физических средах и соответственно с различными протоколами доступа.

Рассмотрим более подробно этот важный вопрос.

1.3 Протоколы доступа. Международные стандарты.

Пакеты и файлы являются основными блоками передачи данных. Обычно они создаются сетевой интерфейсной картой и идентифицированы для каждого типа сети. Кроме имен, адресов и описания данных они содержат коды с обнаружением ошибок для проверки корректности данных после пересылки. IEEE (Institute of Electrical and Electronics Engineers) разработал группу стандартов 802.X, которая покрывает уровень канала данных модели OSI. Структура 802.X разбивает этот уровень канала данных еще на два подуровня:

- подуровень, общий для всех методов доступа к среде LLC (Logical Link Control);
- подуровень доступа к среде (Media Access Control - MAC).

Подуровень LLC известен как спецификация стандарта 802.2.

Подуровень MAC зависит от используемой физической среды и топологии и представлен стандартами 802.3, 802.4 и 802.5.

MAC-адрес - это уникальное число, сохраняемое на каждой сетевой интерфейсной карте. Он позволяет организовать соединения внутри локальной сети, а также используется как часть схемы сетевой адресации.

1.3.1 Метод множественного доступа с контролем несущей и обнаружения столкновений (CSMA/CD(802.3)).

Способ доступа к среде, связанный с Ethernet, известен как CSMA/CD и означает:

Контроль несущей (Carrier Sense - CS) означает, что устройство перед посылкой данных "слушает" состояние ЛВС для того, чтобы выяснить, не осуществляется ли по сети трафик данных. Если сеть свободна, то данные помещаются в сеть.

Множественный доступ (Multiple Access - MA) означает, что любая станция может начать пересылку, когда линия свободна, не ожидая разрешения. Однако, если две станции готовы передать данные в одно и то же время и начинают передачу одновременно, то результатом будет искаженный сигнал.

Обнаружение столкновений (Collision Detection - CD) означает, что обе станции обнаруживают столкновение. Каждая из станций немедленно приостанавливает попытку послать данные и начинает отсчитывать задержку. В произвольный момент станция делает очередную попытку послать данные.

CSMA/CD используется как сетью Ethernet, так и сетями, построенными на базе стандарта 802.3. Ethernet немного отличается от стандарта 802.3, который содержит дополнительную адресную информацию. Однако, 802.3 и Ethernet могут существовать в одной и той же среде, и большинство людей говорят об Ethernet, даже если подразумевают стандарт 802.3.

1.3.2 Кольцо с передачей маркера (802.5)

В системе с передачей маркера рабочая станция, желающая послать данные, должна сначала дождаться получения специального маркера (token). Это значит, что только одна станция может послать данные в любой момент времени, тем самым устраняется возможность коллизии данных.

Маркер представляет собой специальный тип пакета данных, который может быть помещен в сеть только одной станцией в кольце. Такую станцию называют активным монитором. Любая станция в сети может быть активным монитором, но в каждый момент времени может существовать

только один активный монитор, другие станции играют роль резервных мониторов. Если отказывает станция – активный монитор, то с небольшой задержкой активизируется резервный монитор.

Станция – активный монитор начинает кольцо, посылая пустой маркер следующей по кольцу станции. Если эта станция не имеет данных для пересылки, она пропускает маркер следующей станции.

Если при получении пустого маркера станция решает послать данные, она меняет пустой маркер на фрейм, содержащий данные, которые нужно переслать, и адрес получателя этих данных, и запускает этот фрейм в сеть.

Фрейм перемещается по кольцу, пропуская другие станции, тоже готовые к посылке данных.

Станция-получатель читает данные и устанавливает определенный бит во фрейме, подтверждающий получение. Фрейм продолжает движение по кольцу, пока не возвратится к отправителю данного фрейма, который проверяет корректность совершенной пересылки, сравнивая данные во фрейме с теми, которые были посланы, и обнуляет бит "получения". Затем фрейм удаляется из кольца. Если больше нет данных для пересылки, станция освобождает пустой маркер, который может быть использован абонентом. В кольце с маркером любая станция может монополизировать сеть, удерживая пустой маркер и превращая его в фрейм. Однако, ожидая пустой маркер, станция может добавить к проходящему мимо фрейму информацию о том, что ей требуется маркер.

1.3.3 Шина с маркером (802.4)

Основное техническое различие между кольцом с маркером и шиной с маркером заключается в том, что в последней каждое устройство может напрямую адресовать любое другое устройство. Устройство не надо слушать в качестве повторителя для сообщений других устройств.

Протокол автоматизации производства (Manufacturing Automation Protocol – MAP) представляет собой реализацию стандарта 802.4 в промышленности.

Сеть ARCNET, разработанная фирмой Standard Microsystem, является известным недорогим решением, также использующим метод шины с маркером, хотя и неопирающимся на стандарт IEEE 802.4 для маркерной шины. ARCNET использует ThinWire кабель вместо дорогой экранированной витой пары, требуемой стандартом фирмы IBM для Token Ring. ARCNET больше всего подходит для компактных деловых сфер. Скорость передач в ARCNET (2.5 Мб/с) ниже, чем в Token Ring или в CSMA/CD.

1.3.4 Сравнительный анализ стандартов.

Token Ring проектировалась с целью полностью исключить столкновения данных, но при этом требуется некоторый интеллект, введенный в аппаратуру или в программу, чтобы поддержать действия по обслуживанию движения маркера. С ростом размеров сетей проектировщику сети проще оценить производительность Token Ring, чем Ethernet. Разработка Token Ring более сложна, чем сети Ethernet, приходится иметь дело с более дорогой схемой размещения сетевой аппаратуры.

Фирма IBM разработала несколько стандартов Token Ring, включая скорости 4 и 16 Мб/с. Ethernet работает со скоростью 10 Мб/с. Существует много показателей, по которым можно добиваться наивысшей производительности. Протокол CSMA/CD для Ethernet подвергается критике со стороны почитателей Token Ring. Они указывают на коллизии, как на недостаток в CSMA/CD и предполагают, что с повышением загруженности ее эффективность будет падать ввиду роста столкновений данных. Однако, на практике CSMA/CD работает достаточно эффективно. Если сетевой тра-

фик становится слишком интенсивным в CSMA/CD, то эту сеть можно разбить на более мелкие подсети с мостами. Такая техника будет рассмотрена позднее.

CSMA/CD – наиболее распространенный метод доступа к среде, используемый более чем в 50% из всех ЛВС, его поддерживает большее число производителей, чем любой другой. Системы на базе протокола 802.3 используют все три типа медного носителя.

Дополнительно разработаны стандарты на сети CSMA/CD с различными кабелями. К ним относятся:

Коаксиальный 10BASE5 – описывает кабель ThickWire для CSMA/CD. Цифра "10" означает поддерживаемую сетью скорость передачи, в данном случае 10 Мб/с. "BASE" означает, что сеть узкополосная. "5" показывает максимальную длину кабеля в сотнях метров (в данном случае 500 метров).

Коаксиальный 10BASE2 – описывает кабель ThinWire для CSMA/CD, при этом скорость передачи 10 Мб/с с максимальными сегментами кабеля до 200 метров.

Витая пара 10BASET – описывает кабель типа витая пара для CSMA/CD. Стандарт 10BASET, принятый комитетом IEEE в 1990 году, поддерживает 10 Мб/с при использовании кабеля в виде неэкранированной витой пары. Устройства могут быть удалены от центрального узла до 100 метров.

Спецификациям кабеля для Token Ring фирма IBM присвоила номера типов 1 и 3. Тип 1 – это экранированная витая пара (shielded twisted pair – STP), обеспечивающая расстояние между узлами 390 метров при скорости 4 Мб/с и 175 метров при скорости 16 Мб/с. Всего на кольце может располагаться до 256 узлов. Тип 3 – неэкранированная витая пара (UTP), которая при аналогичных скоростях обеспечивает расстояние в 120 и 80 метров соответственно. При этом в кольце может быть до 72 узлов.

Стандарт FDDI предложен для управления доступом к оптоволоконному кабелю. Он устанавливает скорость в 100 Мб/с на двойном кольце протяженностью до 100 км. В кольце может быть до 1000 узлов, а расстояние между узлами доходить до 2000 метров. Для доступа к среде FDDI использует метод временного маркера. Метод позволяет маркеру переносить сообщения более чем от одной станции одновременно, что увеличивает эффективность системы.

1.3.5 Выбор топологии и стандартов.

Приведенный выше анализ топологий сетей и соответствующих им стандартов доступа применительно к проектируемой локальной сети ВЦ РАН с учетом имеющейся базовой среды передачи данных, изложенной выше, приводит к следующим выводам:

1. В основу проектируемой сети первой очереди должна быть положена сеть Ethernet со звездной топологией и стандартом CSMA/CD на неэкранированных витых парах типа 10 BASET.
2. В проектируемой сети должна быть предусмотрена возможность использования коаксиального кабеля по стандарту 10BASE5 (ThickWire) с максимальной длиной кабеля до 500 метров или 10BASE2 (ThinWire) с максимальной длиной до 200 метров. Эта часть проекта переносится на вторую очередь и рассматривается как "расширение в перспективе".
3. Участки локальной сети, соединяющие центральный узел с мощными высокопроизводительными серверами, должны быть спроектированы на стандартах оптоволоконных линий FDDI, допуская при этом возможность работы в первой очереди реализации на стандарте 10BASET.
4. Проектируемая локальная сеть должна иметь возможность выхода на региональную сеть WAN (Wide Area Network) и международную

сеть Internet по двум независимым друг от друга каналам на физическом уровне:

- a) по выделенным и коммутируемым телефонным каналам связи - в первой очереди;
- b) с выходом на оптоволоконную магистральную линию (backbone) в рамках выполнения крупных федеральных проектов и программ по информатизации РФ - в перспективе.

Неотъемлемой частью п. а) является также необходимость предусмотреть в проекте предоставление возможностей доступа к локальной сети ЛВС ВЦ РАН удаленным абонентам по выделенным или коммутируемым телефонным каналам связи.

С выбором топологии, среды передачи данных и методов доступа завершается задача по двум нижним уровням модели OSI - физическому и каналу данных. В нашем проекте сеть имеет явно неоднородную, гетерогенную структуру, что характерно для всех локальных сетей такого масштаба. Однако, для принятия решения по созданию центра коммутации необходимо рассмотреть вкратце еще два уровня модели OSI - сетевой и транспортный, т.к. поддержка этих уровней должна быть обеспечена как с точки зрения программной поддержки, так и в концептуальном плане проекта.

1.4 Сетевой и транспортный уровень модели OSI.

Уровень 3, сетевой уровень, отвечает за соединение любых двух станций в сети. (Это довольно просто в небольшой локальной сети и, как правило, в основном реализуется на втором уровне). Уровень 4, транспортный уровень, обеспечивает, чтобы данные, передаваемые по сети, были доступны третьему уровню. Сообщение может быть представлено в виде отдельных пакетов, проходящих по различным маршрутам. Транспортный уровень контролирует, чтобы пакеты доставлялись адресату в нужном порядке и без ошибок.

Сетевой и транспортный уровни обычно тесно связаны между собой. Данный раздел объяснит некоторые основные характеристики этих уровней и, таким образом, поможет читателям прояснить различия в производительности между некоторыми широко известными протокольными "парами", такими, как TCP/IP и IPX/SPX фирмы Novell.

Слово сеть может подразумевать некоторый набор подсетей или локальных сетей в различных местах, связанных вместе через WAN. Уровень 3 должен знать все адреса и все возможные соединения в сети. Его задача - найти путь от одного компьютера к другому. Этот путь может быть очень сложным, проходящим через несколько других компьютеров. Часто существует более одного пути для того, чтобы попасть из одного места в другое.

1.4.1 TCP/IP

Протокол управления передачей/межсетевой протокол (Transmission Control Protocol/Internet Protocol -TCP/IP) был разработан в начале 70-х. Этот комплект протоколов использовался для объединения в сеть компьютеров университетов и государственных лабораторий, связанных с Управлением перспективных исследований МО США (DARPA).

Хотя он разрабатывался как некоммерческий стандарт для региональных сетей, TCP/IP вскоре был принят другими организациями и предприятиями. Очень популярный в качестве протокола WAN, он приобрел популярность и как протокол локальных сетей.

TCP/IP стал de facto стандартом взаимодействия. Однако, компьютерный мир в 90-х годах осваивает и протоколы OSI. Основная причина выбора новых протоколов заключается том, что для каждого устройства в

"мире" TCP/IP выделяется 32-разрядный уникальный IP адрес. Разработчики TCP/IP не могли знать, что к сегодняшнему дню потребуются миллионы адресов, и выбор ограниченного 32 битами адреса означает, что мы оказываемся за пределами адресного пространства TCP/IP. Адреса распределяются всемирной организацией по запросам, однако их осталось немного, а большие блоки адресов при этом недоступны для использования, поскольку были зарезервированы различными компаниями.

Концепция Ethernet предполагала ее широкое распространение. Каждая сетевая интерфейсная карта (NIC) имела свой собственный 48-битовый MAC адрес, который предоставляет на миллионы адресов больше, чем 32-битовый адрес TCP/IP.

Разработана программная поддержка для разрешения адресных конфликтов. Когда ЛВС получает пакет с IP адресом, она преобразует его в MAC-адрес для соответствующего компьютера в этой ЛВС. Одним из примеров является Протокол разрешения адресов (Address Resolution Protocol - ARP). В TCP/IP для случая, когда на канальном уровне используется Ethernet, между сетевым и канальном уровнями используется ARP - преобразует 32-битный IP-адрес в 48 разрядный Ethernet-адрес и RARP (Reverse ARP) - обратное преобразование.

1.4.2 Производительность транспортного протокола.

Когда сетевые станции обмениваются сообщениями, они должны подтверждать посредством квитирования корректность получаемой информации. Это выполняется через так называемый транспортный кадр, который берет на себя ответственность за пересылку данных и адресов от сетевой операционной системы программам и аппаратуре, которые помещают информацию "на провода".

Простейшее квитирование заключается в том, что принимаемая станция посылает "ОК" отправителю на каждый посланный пакет данных. Прежде чем передать очередной пакет данных, отправитель ожидает прихода этого квитующего сигнала. Если тот не приходит через определенный промежуток времени, или, если возвращенное сообщение указывает на ошибку в передаче, отправитель повторяет пересылку. Такой тип протокола известен под несколькими именами, включая позитивное квитирование с повторением, неоконный протокол и протокол ring-pong.

Для сети, в которой время передачи сообщения между абонентами невелико, например, для ЛВС, такой тип протокола обычно гарантирует быструю и надежную коммутацию данных. Так как такой протокол легко программируется, он выполняется достаточно быстро и требует минимальных компьютерных ресурсов.

На транспортном уровне, наряду с протоколами TCP, обеспечивающими надежность благодаря квитированию, используются и другие протоколы. UDP (User Datagram Protocol) - в отличие от TCP не передает информацию об ошибках, используется для приема/передачи коротких сообщений (например, имен в DNS-сервисе). Третий протокол транспортного уровня, используемый для управления ошибками - ICMP (Internet Control Message Protocol).

Уровни сетевого стандарта TCP/IP хорошо согласуются с моделью OSI. Так как TCP/IP не определяет конкретные протоколы для физического уровня и уровня канала данных модели OSI, он работает с различными стандартами для локальных и региональных сетей. Чаще всего, однако, он используется со стандартами физического уровня и канала данных, определенных в спецификациях IEEE 802 (Token Ring и Ethernet) и X.25 (с коммутацией пакетов).

Межсетевой протокол IP соответствует сетевому уровню 3 модели OSI и, следовательно, отвечает за маршрутизацию данных по сети и между сетями. IP действительно отвечает только за маршрутизацию данных (отправитель, получатель и путь между ними, по которому пересылаются данные).

IP хорошо подходит к региональным сетям, т.к. разработан для обработки сообщений, передаваемых между несколькими компьютерами и по нескольким возможным путям.

Протокол управления передачей TCP имеет дело с 4 функциями транспортного уровня. После того, как IP установил электронное соединение между двумя устройствами, протокол управления передачей обслуживает связь, обеспечивая доставку данных в нужном порядке и с необходимой скоростью. TCP был разработан для WAN, покрывающих тысячи километров, поэтому вопросы надежности учитывались при его проектировании. Он обеспечивает более высокий уровень обнаружения и коррекции ошибок, чем транспортные протоколы типа XNS фирмы Xerox или IPX/SPX фирмы Novell, которые должны обеспечивать скоростные показатели по надежному локальному кабелю.

Протокол TCP/IP обладает возможностью просматривать все возможные пути, что требует определенных затрат, которая делает TCP/IP привлекательным для определенных применений, но не особенно быстрым. Существуют буквально сотни реализаций протокола TCP/IP для систем, охватывающих как суперкомпьютеры Cray, так и компьютеры Commodore Amigas. Большинство производителей компьютеров поддерживают протокол TCP/IP в том или ином виде.

После краткого изложения основных концепций сети, а также выбора стандартов доступа, протоколов сетевых и транспортных уровней рассмотрим вопрос, который является стратегически важным не только для реализации первой очереди проекта, но и для развития научно-технической базы института на ближайшие годы. Таким вопросом является создание коммутационного центра локальной сети с выходом на региональные сети по телефонным и оптоволоконным линиям связи.

1.5 Центр коммутации.

В основу принятия решения при создании такого центра положены следующие основные принципы:

- сеть ЛВС ВЦ РАН есть динамически развиваемая сеть;
- динамика ее развития включает в себя как развитие топологий, так и включение новых вычислительных ресурсов и мощностей;
- центр коммутации должен работать с неоднородной, гетерогенной средой передачи данных, включающей в себя витые пары (УТР), коаксиальный кабель и оптоволоконные линии связи;
- создаваемый центр коммутации должен иметь как минимум 10-кратный запас по производительности с точки зрения выполнения трафиков по сравнению с первой очередью его реализации;
- динамическое развитие сети должно происходить не путем замены аппаратно-программных средств в центре коммутации, а расширением конфигурации или добавлением новых узлов, распределенных в пределах ограниченной территории и совместимых на программно-аппаратном уровне.

В спектре предлагаемых на мировом рынке средств коммуникации с точки зрения минимальных затрат без возможностей использования оптоволоконных каналов предлагается строить сеть по принципу модульного расширения на концентраторах каналов HUB, которые связаны между собой с помощью дополнительных устройств-мостов (Bridge). Помимо отсутствия в этом варианте оптоволоконных средств связи, существенный недостаток в нем заключается в дополнительных нагрузках на Bridge при наращивании сети с помощью дополнительных кластеров. При этом нагрузка увеличивается непропорционально числу кластеров. Исключая этот вариант, как неприемлемый для центрального узла, он может быть рекомендован для отдельных участков сети.

Следующий вариант имеет в своей архитектуре также три независимых кластера. Связь с серверами - рабочими станциями может быть реализована по оптоволоконным линиям связи, если последние снабжены

адаптерами типа FDDI - SBUS - module. Связь между тремя кластерами осуществляется с помощью NAE (Netware Application Engine (Novell)) с одной дополнительной LAN-картой вместо мостов. Устройство известно под названием Neth Worth Series 3000, которое имеет шести/восьмислотовую конструкцию, один модуль управления (Management module), а также один шестиканальный оптоволоконный модуль и блок независимого источника питания.

В качестве альтернативы, наиболее близкой к нашим требованиям на первом этапе, может быть рассмотрен вариант 3 с использованием устройства фирмы 3Com (США) Linkbuilder MSH (Multi Services Hub). Данное устройство - одно из самых распространенных в мире локальных сетей. Построенное по модульному принципу на слотовой конструкции, устройство включает в состав те же функциональные модули, что и вариант 2, со значительными расширениями как в программной, так и в аппаратной части. С помощью цепочки устройств типа MSH, а также сетевых компонент типа Netbuilder II и Bridge Router может быть, например, построена сеть, которая включает в себя различные топологии на отдельных участках:

- FDDI Backbone;
- Token Ring 16 Мб/с;
- Token Ring 4 Мб/с;
- 10BaseT.

Однако, несмотря на представленную здесь гибкость и многофункциональность системы, данное устройство уже не отвечает новейшим современным требованиям в области сетевых исследований.

На мировом рынке уже появились новейшие суперсовременные средства коммуникации для локальных сетей, имеющие производительность до 400.000 пакетов/сек и пропускную способность через общую шину типа VME до 300-400 Мб/с.

Из широкого спектра предлагаемых на рынке сетевых устройств авторы проекта остановились на двух альтернативных вариантах выбора, равнозначных по техническим характеристикам:

- A. ALANTEC 3220 Intelligent SNMP Power HUB/Router;
- B. Linkbuilder 3GH (Third - Generation HUB)

Функциональные и предельные характеристики указанных устройств практически одинаковы. Предпочтение варианту B - Linkbuilder 3GH - дано по следующим критериям:

1. ценовые характеристики оборудования фирмы 3Com выигрывают по сравнению с ALANTEC 3220;
2. комплектация ALANTECа для развертывания локальной сети включает в качестве обязательных компонент собственные стандартные кабели UTP (STP) и AUI в качестве гаранта работоспособности сети. По существу это обстоятельство стало решающим фактором при решении вопроса в пользу 3GH, поскольку к началу проектных и пуско-наладочных работ сеть передачи данных на витых парах (UTP) была уже развернута на ответственных кабелях. Следует заметить, что авторы проекта при этом взяли на себя всю ответственность за установку сети на основании проведенных предварительно инженерных исследований, так как поставщик сетевого оборудования фирма COMPU-SHACK (Германия), не имея результатов испытаний среды передачи данных ВЦ РАН, не гарантировала успеха установки сети в целом.
3. Linkbuilder 3GH, с нашей точки зрения, имеет максимально гибкий функциональный набор модулей для развертывания в будущем оптоволоконной среды передачи данных, который обеспечивается наличием в архитектуре устройства модуля стыковки с магистральной оптоволоконной линией связи (backbone) FBCM и концентратора FCM, что будет рассмотрено ниже.

Завершая сравнительный анализ вариантов построения центра коммутации ЛВС ВЦ РАН, представим общую архитектуру сети, которая должна была стать базовой при условии получения разрешения и соответствующих

лицензий КОКОМа на компоненты сети, в первую очередь в части оптоволоконных линий связи и соответствующих адаптеров.

Так как лицензирование такого проекта ЛВС ВЦ РАН к началу выполнения работ в 1992-1993 гг. было под вопросом, что подтвердилось впоследствии, полностью оправдала себя концепция поэтапного развития, положенная в основу проекта. Она позволила не только реализовать развертывание локальной сети по принципу постепенного наращивания как серверного уровня, так и среды передачи данных, но и более эффективно планировать капиталовложения с учетом постепенного снижения цен на мировом рынке на компьютерную технику и средства связи.

Рассмотрим общую конфигурацию ЛВ 3GH. Данное устройство представляет из себя 12-слотовую конструкцию, которая устанавливается в соответствии с техническими условиями в отдельный шкаф-модуль, как составная часть сетевого комплекса. Устройство имеет модульную архитектуру и может быть скомплектовано из следующих функционально независимых плат-модулей:

1. Блока источника питания (Linkbuilder 3GH Dual Power Supply 3C 9001 - ME) - обязательная часть;
2. Модуля системного управления (Linkbuilder 3GH Enhanced System Management Module 3C 9020) - SMM - обязательная часть;
3. Модуль сети Ethernet (Linkbuilder 3GH Ethernet LAN Module) - ELM следующих модификаций:
 - A. А. ELM с выходом на 8 портов для витых пар (ELM 10BASET);
 - B. В. ELM с выходом на интерфейс АUI по стандарту IEEE 802.3 и скомплектованный согласующим устройством - трансивером MAU;
 - C. ELM с выходом на 10BASE5 (Thick-net);
 - D. ELM с выходом на 10BASE2 (Thin-net);
 - E. ELM с выходом на удаленные мультиплексоры HUB (Non - 10BASET, UTP HUB);
 - F. ELM (ESM) - Switch-модуль для организации групповых трафиков по принципу мостов (bridges);
4. Модуля подключения к магистральному оптоволоконному кабелю (FDDI Backbone Connectivity Module) - FBCM;
5. Модуля оптоволоконного концентратора (FDDI Concentrator Module) - FCM.

Представленный спектр модулей, а также возможности построения сложнейших структур путем соединения самих устройств ЛВ 3GH в цепочку по различным топологиям предоставляет практически неограниченные возможности для построения сетей.

Минимальная комплектация ЛВ 3GH должна включать в себя один модуль системного менеджера SMM любой модификации (4.X или 5.X), один из модулей ELM.X в соответствии с реализуемым вариантом проекта или один модуль концентратора FCM.

В плане перспективного развития локальной сети ВЦ РАН и стыковки ее с другими локальными сетями (ИОХ РАН, ИРЭ РАН, МИ РАН, МГУ и т.п.) должны быть предусмотрены специализированные устройства мосты (bridges). Эта проблема в данном случае решается добавлением в конфигурацию специального функционального блока ESM (Ethernet Switching Module), который представляет одну из модификаций ELM - модуля.

Эта же проблема может быть решена и другим путем: например, с помощью удаленного интеллектуального мультиплексора HUB 10BASET, вынесенного на максимальное удаление от места установки ЛВ 3GH. Практический интерес представляют также возможности данного устройства для развертывания локальной сети Ethernet только на собственных модулях ELM.X. В этом крайнем случае максимальное количество подключаемых РС и рабочих станций по витым парам равно 88, а максимальная скорость передачи составляет 110 Мбит/сек через общую шину VME. От увеличения количества подключаемых к сети рабочих станций посредством выносных HUBов - максимумо 11x8x12=1056, общая производительность системы не

изменится, так как количество логических сегментов - кластеров остается неизменным (11x10 BASE T). Максимальная производительность системы на шине VME составляет 3x100 Мб/с.

Поэтому можно представить другой крайний вариант комплектации сетевой системы на LB 3GH:

- один модуль SMM;
- 11 модулей оптоволоконных концентраторов FCM.

В этом случае имеется возможность стыковки с ETHERNET:

- a) 11x4=44 рабочих станций с одинарным входом (SAS);
- b) 11x2=22 рабочих станций с двойным входом (DAS) с соответствующими FDDI-S-BUS картами со стороны рабочих станций в обоих вариантах а и в.

В реальных условиях комплектация LB 3GH наращивается динамически в пределах двух указанных крайних случаев с одним исключительной важности дополнением - наличием в его составе хотя бы одного модуля FBCM для стыковки с оптоволоконной магистральной линией связи (backbone).

Изложенные здесь принципы заполнения ЛВС ВЦ РАН в части оптоволоконной среды входят во вторую очередь проекта и возможны только при выполнении следующих неперенных условий:

- снятие ограничений КОКОМом;
- наличие финансовых возможностей, в первую очередь валютных средств;
- возрастающая потребность со стороны пользовательской среды в мощных серверах, вычислительных ресурсах с адекватной средой передачи данных.

1.6 Физическая среда передачи на витых парах. Исследования и реализация.

Вопрос создания физической среды передачи данных при построении локальных сетей и региональных сетей - один из важнейших вопросов, который во многом определяет успех при выполнении подобных проектов. Практика построения локальных сетей в 80-е годы базировалась преимущественно на "тонком" Ethernet (Thinlink), краткое описание которого дано выше. В основу физической среды при этой топологии положен коаксиальный кабель с архитектурой общей шины. Однако, имея несомненное преимущество по помехозащищенности и экономичности первоначального капиталовложения, такая сеть мало пригодна и бесперспективна для построения постоянно развивающихся открытых сетей для крупных исследовательских центров. Крупные телефонные компании AT&T, Cabletron Systems, Fore Systems и др., имеющие богатейший опыт развертывания сетей, предлагают продукт, который называется Структурированная Кабельная Сеть. В это понятие помимо конструктивных элементов входит также инженерная проработка проекта, включая настройку архитектуры сети на особенности здания и режима использования.

Принципиальная особенность структуры кабельных сетей заключается в том, что тщательно спроектированные и удачно настроенные сети гарантируются в течение 15-20 лет даже при возникновении новейших технологий на уровне серверов и конечных абонентских точек. Фактор "времени жизни" структурированных сетей - одна из важнейших характеристик кабельных сетей, так как сроки 1-2 года определяют текущие расходы, а 10-15 лет - это уже капиталовложения, соизмеримые с постройкой нового здания или капитальным ремонтом.

В сентябре 1986 г. на научной конференции Ассоциации пользователей БЭСМ-6 в г. Киеве, организованной институтом кибернетики им. В.М.Глушкова АНУССР, был представлен доклад: "Структурированная сеть ВЦ АН СССР" (Горбатов Э.К., Михайлов Г.М., Привезенцев Ю.А.), кото-

рый, к сожалению, не был опубликован своевременно из-за финансовых трудностей, возникших на Украине в связи с Чернобыльской аварией.

Необходимость создания такой локальной сети для ВЦ АН СССР была актуальной задачей еще в 80-е годы, так как в институте была развернута одна из самых передовых по тому времени систем коллективного пользования, позволяющая осуществлять удаленный доступ к вычислительным ресурсам 128 абонентов и более внутри и вне здания ВЦ АН СССР. Данная сеть работала в составе общей системы коллективного пользования до 1992 г. и практически в своей физической основе без каких-либо изменений стала базовой сетью для развертывания локальной сети ЛВС ВЦ РАН в течение 1992-1993 гг. Важнейшим фактором успеха при создании новой структурированной сети авторы считают принцип 100% отказа от старых сетей, заложенных в здание в 50-е годы и переход на новые стандарты телефонных кабелей на витых парах. По классификации AT&T различают два уровня по числу витков на единицу длины:

- уровень 3 (Level 3 - Voice Grade);
- уровень 5 (Level 5 - Data Grade).

Уровень 3 сертифицирован для использования в сетях со скоростями передачи до 16 Мб/с, а уровень 5 - до 100 Мб/с. Неотъемлемой частью структурированных сетей являются коммутационные панели-"кроссы", телефонные распределительные коробки и телефонные розетки, выполненные по международным стандартам.

Сеть развернута на телефонном кабеле отечественного производства типа ТПП парной скрутки с шагом от 100 до 600 мм различной емкости:

- 10x2x0.4;
- 20x2x0.4;
- 30x2x0.4;
- 100x2x0.4.

Кроссы выполнены на шкафах типа ШРП-1200 и сосредоточены в одном узле. В этот же узел имеют вход общегородские телефонные кабели с АТС 135, 137 и 939, выполненные на кабелях 2x100x2x0.4 и 300x2x0.64.

Наши исследования, проведенные после развертывания сети на объекте ВЦ РАН до начала реализации проекта "ЛВС ВЦ РАН", дали следующие результаты:

1. Длины лучей звездной топологии имеют диапазон
 - $L_{min} = 43 \text{ м,}$
 - $L_{max} = 155 \text{ м;}$
2. Активная составляющая сопротивления линий находится в диапазоне
 - $R_{min} = 14 \text{ ом,}$
 - $R_{max} = 42 \text{ ом;}$
3. распределенная емкость линий, т.е. каждой витой пары, имеет величины
 - $C_{min} = 2.6 \text{ нФ,}$
 - $C_{max} = 8.1 \text{ нФ;}$
4. Волновое сопротивление линий (импеданс)
 - $Z < 80 \text{ ом.}$

Структурированная кабельная сеть ВЦ РАН в своей архитектуре имеет центральный узел коммутации, где помимо кросспанелей находится также активный элемент - центр управления сетью Linkbuilder. Кроме того, архитектура сети включает в себя ряд локальных панелей коммутации, позволяющих развивать "дерево" сети с включением дополнительных активных элементов - HUBов.

Другой важнейшей особенностью коммуникационной сети ВЦ РАН является ее многофункциональность. Она обеспечивает не только компьютерную сеть, но и множество других - телефонные, радио, охранные, противопожарные и другие. В этом смысле сеть универсальна и имеет значительную избыточность линий, позволяет без каких-либо изменений топологии сети создавать новые рабочие места.

В заключение следует заметить, что скорость передачи 10 Мб/с по витым парам типа 10BASET безусловно накладывает ограничения на общую пропускную способность локальной сети. Наиболее критичными при этом оказываются участки или каналы связи, соединяющие серверы с центральным узлом Linkbuilder. Среди всех существующих вариантов, описанных в данной главе, нельзя исключать возможность использования Ethernet на витых парах для 100 Мб/с, для чего эти участки-лучи должны удовлетворять требованиям стандарта Level 5 при соответствующей программной поддержке и новых протоколов.

2. Рабочие станции фирмы SUN Microsystems.

2.1 Аппаратные особенности рабочих станций фирмы SUN.

Одним из наиболее мощных производителей рабочих станций в мире является фирма SUN Microsystems (США). По оценкам фирмы Dataquest более 74% процентов рабочих станций, установленных в мире на начало 1992 г., составляют рабочие станции SUN и фирм, выпускающих станции по ее лицензиям, причем, количество установленных фирмой рабочих станций в течение последних лет ежегодно почти удваивалось и к началу 1992 г. составило 450 тыс. систем.

Фирма в течение долгого времени выпускала рабочие станции на основе 32-х разрядного процессора серии 68К фирмы MOTOROLA. Однако, свое ведущее положение SUN заняла после освоения выпуска станций серии SPARC, уже упоминавшихся выше. Особенностью архитектуры SPARC является практически 100% сбалансированность всех узлов ЭВМ, что за счет отсутствия "узких мест" в системе обеспечивает возможность пропорционального увеличения быстродействия всех узлов системы и увеличение производительности с тактовой частотой по мере роста технологических возможностей.

Это свойство архитектуры позволяет сохранить полную программную совместимость всех SPARC процессоров при переходе от ранних, относительно медленных моделей к последующим. По оценке инженеров фирмы, SPARC-процессор в состоянии обеспечить в перспективе производительность до 1 млрд. операций в секунду и более. SPARC архитектура относится к архитектурам типа RISC. Основным достоинством этой архитектуры является то, что при увеличении тактовой частоты процессора пропорционально возрастает скорость выполнения всех команд, а, следовательно, при росте быстродействия строго пропорционально возрастает и производительность системы, в отличие от ЭВМ с полным набором команд (CISC). Это хорошо заметно на эволюции рабочих станций серии SPARC в течение последних четырех лет.

Сегодня номенклатура изделий SUN включает большое разнообразие машин с широким спектром возможностей и удовлетворяющих самых различных пользователей как по техническим характеристикам, так и по стоимости. Сюда входят и IBM - совместимые персональные ЭВМ SUN i386, которые представляют системы нижнего уровня, и наиболее мощные ЭВМ, выполненные на основе SPARC процессоров - многопроцессорные SPARC серверы серии 690MP, производительность которых достигает 100 млн. операций в секунду и более. Эти серверы используются в качестве файловых серверов, серверов баз данных и прикладных серверов в распределенных вычислительных системах. Производительность серверов может быть значительно увеличена за счет установки дополнительных векторных процессоров с быстродействием 256 млн. операций с плавающей точкой в секунду.

Наибольший объем занимают рабочие станции серии SPARC. Рабочие станции отличаются как по своему быстродействию, так и по составу используемых дополнительных узлов, обеспечивающих решение самых различ-

ных прикладных задач при широком диапазоне стоимости, способном удовлетворить практически любого пользователя.

Рабочая станция SPARCstation ELC представляет собой станцию с наименьшей стоимостью, не превосходящей стоимости ПЭВМ. При этом технические характеристики станции существенно выше, чем даже у наиболее мощных ПЭВМ, таких как IBM PC 486. SPARC-процессор станции имеет тактовую частоту 33 мгц и производительность около 24 млн. операций в секунду (MIPS).

В базовом комплекте станция имеет ОЗУ объемом 8 МБ, расширяемое до 64 МБ. Емкость дискового накопителя равна 207 МБ и может быть расширена до 5,3 ГБ. Станция имеет встроенный SCSI контроллер для подключения внешних устройств, встроенный контроллер сети Ethernet, два последовательных порта RS-232 и порт ввода/вывода audio информации. Станция оснащается 17" черно-белым монитором с разрешением 1152x900 точек, который обеспечивает вывод полутоновых изображений. Выпускается бездисковый вариант станции SPARCstation SLC, предназначенный для работы в локальной сети с загрузкой через сеть из сетевого сервера. Стоимость такого варианта приблизительно вдвое ниже стоимости станции ELC.

Следующей по возможностям и стоимости является станция SPARCstation IPC. Эта станция уже может комплектоваться по выбору черно-белым или цветным монитором. Производительность этой станции несколько ниже, чем у ELC (18 MIPS при тактовой частоте 25 мгц).

Однако эта станция допускает значительно большее разнообразие применений и соответственно большее количество подключаемых внешних устройств и применяемого программного обеспечения. В базовом комплекте станция имеет ОЗУ объемом 8 МБ, расширяемое до 48 МБ, встроенный дисковый накопитель емкостью 207 МБ, расширяемый до 15,6 ГБ. Кроме этого имеется встроенный дисковод для 3,5" гибких дисков, емкость в 1,44 МБ, совместимый по формату с аналогичными накопителями, используемыми в IBM PC. В зависимости от пожеланий потребителя, станция может быть укомплектована 17" черно-белым монитором или же цветным монитором 16" или 19". Станция имеет встроенный SCSI контроллер, встроенный Ethernet контроллер, два порта RS-232 и audio порт. Благодаря наличию возможности расширения дискового накопителя, станция может быть использована в качестве файлового сервера в локальной сети. Для расширения аппаратуры станции предусмотрено два свободных разъема на внутренней магистрали SBus.

Рабочая станция SPARCstation IPX уже использует SPARC процессор с тактовой частотой 40 мгц. Производительность этой станции оценивается уже в 28,5 MIPS. В отличие от предыдущих станций станция IPX уже оснащена встроенным графическим ускорителем, который обеспечивает генерацию цветных или полутоновых черно-белых изображений со скоростью 300 тыс. 3D или 400 тыс. 2D векторов в секунду. Эта станция является младшей моделью станций, использующих аппаратно реализованную графику.

Рабочие станции SPARC имеют несколько типов графических ускорителей, обеспечивающих аппаратную реализацию ряда стандартных функций. Об этом подробнее будет сказано ниже. Базовая конфигурация станции имеет ОЗУ емкостью 16 МБ, расширяемое до 64 МБ. Имеется встроенный дисковый накопитель емкостью 207 или 427 МБ. За счет присоединения внешних накопителей емкость дисков может быть увеличена до 15,6 ГБ. В базовой конфигурации также имеется 3,5" гибкий дисковод, SCSI, Ethernet контроллеры, два последовательных порта RS-232 и audio порт. Станция может оснащаться 17" черно-белым или 16" и 19" цветным монитором. Разрешение для всех мониторов 1152x900 точек.

Ряд следующих станций, которые имеют общее наименование SPARCstation 2, построены на основе одинакового процессора (аналогичного процессору станции IPX) и отличаются, главным образом, типом применяемого аппаратного графического ускорителя. При тактовой частоте 40 мгц производительность процессора составляет 28,5 MIPS. В базовом

вом комплекте станции имеется ОЗУ с начальной емкостью 32 МБ. ОЗУ может быть расширено до 128 МБ. Встроенный диск в базовой конфигурации 424 МБ и за счет дополнительных внешних накопителей емкость внешней памяти можно расширять до 20,8 МБ. Также как и другие станции SPARCstation 2 имеет встроенный 3,5" накопитель на гибком диске. Имеется также 3 дополнительных SBus разъема для включения добавочных устройств и встроенные SCSI, Ethernet контроллеры, два порта RS-232 и audio порт. Станции могут использовать как черно-белый 17", так и 16" и 19" (21" для SPARCstation 2GT) цветной монитор. Кроме основного монитора допускается подключение дополнительного.

Различия между графическими адаптерами в значительной степени устраняются с помощью существующих графических библиотек. Естественно, что в случае, когда те или иные функции не поддерживаются аппаратно, это влечет за собой снижение скорости вывода. Однако наличие сквозной аппаратно-программной совместимости в рамках стандартной графики фирмы обеспечивает полноценную мобильность программных продуктов между всеми станциями семейства SPARCstation 2.

Программные пакеты: SunPHIGS, SunGKS и XGL полностью используют аппаратные средства адаптеров GX. Оконный сервер X11/NeWS также реализует преимущества, даваемые ускорителем GX при выполнении графических функций системами Xlib и PostScript. Точно также ускоряются графические операции в системе Pixwin/Pixrect.

2.2 Программная среда рабочих станций и ЭВМ фирмы SUN.

Стандартной операционной системой ЭВМ фирмы SUN является SunOS. SunOS - это фирменная версия ОС UNIX System V AT&T с расширениями BSD 4.3 (Berkeley Software Distribution). SunOS применяется на всех трех основных архитектурах, используемых фирмой: SPARC, Motorola 68K и i80X86. SunOS ориентирована на такие аппаратные стандарты, как SBus, SCSI, VMEbus и Ethernet. ОС содержит богатый набор программных механизмов: "STREAMS", "sockets", файлы с отображением в память, асинхронный ввод/вывод и т.д.

Наличие этих механизмов обеспечивает использование доступа к информации и удаленным процессам через сеть, применение "оконных" интерфейсов, работу с графикой. SunOS является программной основой для разработчиков программного обеспечения. Стремление к реализации открытой операционной среды, обеспечивающей максимальную мобильность прикладных программ, привело к созданию программных компонент, удовлетворяющих основным промышленным стандартам: POSIX, X/OPEN Portability Guide и ряду других. Сейчас проводится работа по созданию версии SunOS, полностью удовлетворяющей спецификациям унифицированного стандарта UNIX System V Release 4 (SVR4).

Рассмотрим некоторые из программных механизмов, упомянутых выше. К таковым относятся прежде всего: RPC (Remote Procedure Call)-механизм запуска процедуры и XDR (eXternal Data Representation)-стандарт для машинно-независимого представления данных. Эти компоненты являются основой для реализации взаимодействия между портами сети на уровнях выше 4-го (транспортного). Этот принцип взаимодействия получил название "клиент-сервер". Отображение файлов в памяти (Mapped files) позволяет процессам обращаться к данным, содержащимся в различных файлах, как к ячейкам памяти в своем адресном пространстве. Этот механизм обеспечивает также разделение доступа к таким файлам различными процессами.

Асинхронный ввод/вывод - это возможность запуска нового процесса ввода/вывода до завершения текущего ввода/вывода. При такой организации прикладные процессы и текущие процессы ввода/вывода происходят одновременно и синхронизируются после завершения обменов. Использование этой возможности позволяет увеличить производительность системы в тех случаях, когда время исполнения прикладной программы явля-

ется критическим параметром. Эта возможность полезна и для прикладных программ, интенсивно работающих с базами данных.

Разделяемая память – это неявный способ межпроцессного взаимодействия. Он является чрезвычайно эффективным способом передачи сообщений от процесса к процессу. Для организации такого способа связи необходимо, чтобы связанные таким образом процессы исполнялись одним и тем же процессором. Для организации межпроцессной связи в том случае, если процессы используют различные процессоры, используются другие способы связи, например, разделяемые файлы. Для обеспечения совместимости со стандартами в SunOS реализованы и стандартные для System V функции, такие как сообщения и семафоры.

Ряд механизмов, реализованных в SunOS, обеспечивает доступ к файловой системе в различных приложениях. Одним из способов доступа к удаленным файлам в локальной сети является система NFS (Network File System). NFS обеспечивает "прозрачный" доступ к файлам в распределенной файловой системе через локальную сеть, даже в случае, если сеть обслуживает неоднородную вычислительную среду. Система NFS является промышленным стандартом и входит в группу, объединяемую стандартом POSIX 1003.8. Программный продукт PC-NFS, обеспечивающий доступ к файлам в среде MS-DOS и SunOS, является собственным продуктом SUN Microsystems.

С начала 1992 г. фирма начала поставки операционной системы SOLARIS. Эта ОС включает в себя уже существующую SunOS 4.1, ряд программных продуктов, ранее поставлявшихся отдельно. В состав SOLARIS, кроме SunOS, входят пакеты OPEN LOOK, OPEN Windows, ONC и DeskSet. С включением в состав ОС системы OPEN LOOK она становится полностью отвечающей стандарту на реализацию пользовательского оконного интерфейса GUI. Система ONC обеспечивает распределенные вычисления в неоднородной локальной сети. DeskSet – это пакет программных продуктов, обеспечивающих работу прикладных программ в среде OpenWindows.

Ранее SUN выпускала набор графических адаптеров CXP, GXP GXI и ТААС. Новое поколение графических адаптеров SUN включает адаптеры (ускорители) GX, GXplus, VX/MVX, GS и GT.

Начиная с версии SunOS 4.1, SUN прекращает поддержку пакетов SunCGI и SunCORE, которые заменяются новыми более мощными графическими стандартами GKS, PHIGS и XLib. SUN осуществляет переход от системы SunView к системе OpenWindows в качестве оконной системы, в среде которой работают графические приложения.

Разнообразие существующих графических интерфейсов обусловлено разнообразием возможных приложений графических систем, включающих графику общего назначения, автоматизацию конторской деятельности, автоматизацию проектирования и полную визуализацию.

Одним из существенных факторов при выборе графического интерфейса является необходимость использования двух или трехмерной графики. Ряд пакетов, таких как SunGKS, Xlib и язык PostScript, обеспечивают двухмерную (2D) графику. SunPHIGS, XGL и SunVision обеспечивают и трехмерную (3D) графику. Все графические интерфейсы могут работать как с цветными, так и с черно-белыми полутонными дисплеями.

Display List (DL) – это механизм, который обеспечивает режим непосредственного отображения графической картинка. По своей сути – это буфер графических команд, который используется непосредственно для перерисовки содержимого экрана и таким образом уменьшает потери времени на передачу графических команд в графический процессор из основного процессора. Графика, использующая эту технику, особенно полезна в тех случаях, когда необходимо отображение большого количества графических данных, которые часто обновляются. Хранение данных в DL экономит время, необходимое на перерисовку всей картинка в тех случаях, когда требуется обновить данные или если происходит искажение данного окна.

Как SunPHIGS, так и SunGKS включают механизм DL. SunPHIGS к тому же использует редактируемый DL, который имеет иерархическую структуру.

Некоторые прикладные программы должны отображать графические примитивы прямо в непосредственном режиме. Техника DL порождает значительные перегрузки, которых не бывает при использовании непосредственного режима.

3. Вычислительная система PARSYTEC GCel 1/64.

В качестве одного из вычислительных серверов в ЛВС ВЦ РАН используется высокопроизводительная система для параллельной обработки фирмы PARSYTEC под названием GIGA CUBE ENTRY LEVEL 1/64 (GCel 1/64).

GCel 1/64 является мощным компьютером типа MIMD (Multiple Instructions Multiple Data) и представляет собой фактически сеть высокопроизводительных микропроцессоров (транспьютеров), каждый из которых имеет собственную память. Между процессорами можно производить обмены "память-память" с достаточно высокой скоростью. GCel 1/64 - это конструктивно один куб. Предусмотрено наращивание мощности системы за счет увеличения числа таких кубов. Максимальное число кубов в системе 256, что эквивалентно 16384 рабочим транспьютерам.

GCel работает в связке с внешней или управляющей машиной (frontend computer). GCel 1/64 не является узлом сети ЛВС ВЦ РАН. Выход на GCel с любого компьютера локальной сети осуществляется через внешнюю машину, которая, будучи узлом сети, обеспечивает такую возможность. Внешней машиной в нашем случае является ЭВМ HAMSTATION-2 (типа SPARCStation - 2). Она обеспечивает интерфейс с пользователем GCel, функции загрузки процессорных узлов GCel и управления им, слежения за работой GCel (monitoring), реализует доступ к файловой системе.

Системное программное обеспечение называется PARIX (PARAllel unIX).

Рассмотрим возможности GCel и PARIX более подробно.

3.1 Аппаратные особенности архитектуры GCel 1/64.

GCel 1/64 - наименьшая аппаратная конструкция, которая может существовать самостоятельно. Будем для краткости называть ее Giga Cube (GC). GC имеет собственный конструктив, в котором размещается основное оборудование, а также система жизнеобеспечения и снабжения электропитанием.

GC состоит из 4-х кластеров. Кластер - это своеобразный атомарный конструктивный блок. Каждый кластер содержит 17 процессорных узлов. Из них 16 основных рабочих и один дополнительный для повышения надежности системы, вступающий в работу в случае отказа любого из основных рабочих узлов. Процессорные узлы размещаются на специальной плате (processor board). Кластер состоит из 2-х таких плат с процессорными узлами.

Каждый GC имеет еще один процессорный узел, управляющий. Если система имеет больше одного GC, то все управляющие узлы связываются между собой, образуя управляющую сеть. Назначение управляющих узлов GC: конфигурирование, начальная установка (reset), слежение за работой системы (мониторинг) и т.д. Таким образом, на GC имеется 69 процессорных узлов.

Процессорный узел содержит транспьютер, главную память объемом 4 Мб, специальное логическое устройство для коррекции одиночных ошибок памяти и выдачи сигнала об отказе при двойных ошибках (EDC-

устройство), а также микросхемы, связанные с датчиками температуры и напряжения в сети.

В качестве процессора процессорного узла используется транспьютер T805 фирмы INMOS. В будущем фирма PARSYTEC предполагала заменить существующие узлы на узлы с транспьютерами T9000 в качестве процессоров. Пока эти намерения остались не реализованными и по последним данным фирма собирается строить процессорные узлы с использованием процессора POWER PC.

Транспьютер T805 является разновидностью RISC процессора. Это однокристалльная интегральная схема с собственной памятью в 4 Кб (помимо главной - объемом 4Мб). T805 имеет основной 32 разрядный процессор и процессор плавающей арифметики, работающий с 64 разрядными числами. Возможна прямая адресация к памяти объемом до 4Gb. В нашей установке главная память транспьютерного узла 4 Мб.

Подобно другим RISC архитектурам T805 имеет специфическую систему команд. Код команды имеет либо 4 бита, либо находится в специальном регистре. Операнды и результат операции размещается в стеке из 3-х регистров. Команды не содержат обращения к памяти, кроме 2-х операций (записи и чтения из памяти). Транспьютер имеет специальные средства для поддержания мультипрограммирования на одном транспьютере (управление очередями, таймеры, приоритеты для процессов). Развернутой системы прерываний транспьютеры не имеют, но имеются средства реакции на некоторые внешние события - т.н. SYSTEM SERVICES - для приведения системы в исходное состояние (reset), проведения начальной загрузки транспьютера, получения информации об ошибках. Специальных средств защиты и управления памятью T805 не имеет, что в некотором смысле оправдано, поскольку принятая на транспьютерах идеология параллельных вычислений предполагает, что параллелизм достигается в основном работой на нескольких транспьютерах.

Многозадачность возможна, но она реализуется таким образом, что на одном транспьютере не могут работать процессы, принадлежащие разным задачам. На одном транспьютере работают в мультипрограммном режиме только процессы пользователя (основной и подчиненные в виде т.н. thread) и служебные процессы их обслуживающие. Защита обеспечивается программным образом за счет жесткого распределения памяти при начальной загрузке программы. Следствием этого является то, что обычный пользователь не программирует в кодах транспьютера и на ассемблере. Важная часть работы T805 - это организация обменов "память-память" между транспьютерами. Каждый транспьютер имеет 4 специальных канала - линка для такого обмена. Каждый линк имеет 2 подканала - для ввода и для вывода. На каждом транспьютере информация в линк (или из линка) передается через специальный линковый интерфейс. Максимальная скорость обмена по линку 20Мб/сек (2.35Мбайт/сек.).

Для реализации логического уровня связи между транспьютерами используется специальное программно настраиваемое устройство - канальный переключатель C004 (INMOS C004 LINK CROSSBAR SWITCH). На физическом уровне каждый транспьютер без использования C004 может быть максимально связан только с 4-мя другими транспьютерами. На логическом уровне, благодаря использованию C004, при соответствующей программной настройке возможно иметь практически неограниченное число виртуальных логических каналов. Для повышения эффективности работы по виртуальным каналам в будущих разработках PARSYTEC предполагал использование новых канальных переключателей C104. В GCel на каждом кластере имеется 4 C004 и каждый из 4-х физических линков транспьютера выходит на разные C004. C004 поддерживает скорость 20 Мб/с.

Физически обмены по линкам происходят синхронно. Это значит, что обмен происходит только тогда, когда на одном транспьютере выставлена команда ввода (или вывода), а на другом соответственно команда вывода (или ввода). Обмен начинается, когда выдана последняя в этой паре команда. В будущих разработках предполагается реализация на физическом уровне асинхронных обменов. В рамках PARIX на языках выс-

кого уровня имеются процедуры асинхронного обмена, но они реализуются средствами эмуляции.

Как уже говорилось выше, из транспьютерных узлов строятся кластеры, 4 кластера составляют куб (GCel). На каждом кубе имеется еще один транспьютерный узел для управления. Этот управляющий узел предназначен для:

- производства начальных установок для куба (процедура reset);
- отслеживание внешней среды через реакцию на сигналы от температурных датчиков и датчиков уровня напряжения,
- конфигурирование связей между узлами через загрузку канальных переключателей (C004) на каждом кластере.

Если система имеет несколько кубов (максимально 256), то управляющие процессорные узлы образуют управляющую сеть.

Ввод-вывод на GC осуществляется с каждого транспьютерного узла по обычным линкам. В рамках PARIX определены конкретные логические линки для ввода-вывода на каждом узле. Информация ввода-вывода передается между GC и внешней машиной по специальным линиям связи, называемым ENTRY USER или просто ENTRY. Количество таких ENTRY может быть различным и определяется в частности специальным интерфейсом, устанавливаемым на материнскую плату внешней машины. И нашем случае мы имеем интерфейс ВВК-S4, который имеет 3 ENTRY (1,2,3) и еще одна линия (0), связывающая внешнюю машину с управляющим узлом GC. Ниже мы увидим, что количество ENTRY для PARIX определяет количество одновременно решаемых задач на GC. По каналам ENTRY осуществляются операции ввода-вывода между GC и периферийными устройствами рабочих станций ЛВС ВЦ РАН.

В заключении этого раздела несколько слов о производительности вычислительной системы GCel 1/64. Физические характеристики GCel 1/64 следующие. Максимальная производительность каждого процессорного узла с транспьютером T805 (30Мгц) оценивается разработчиком в 4.3 MFLOPS и 30 MIPS, максимальная скорость обмена между узлами по линкам 20Мбит/сек (2.35 Мбайт/сек). Это значит, что максимальная производительность для GCel 1/64 может быть 0.28 GFLOPS и 1.9 GMIPS. Реальная производительность GCel 1/64, естественно, меньше, она зависит от характера конкретной задачи, от качества распараллеливания данного алгоритма, от объема и характера обменов по линкам, от качества системного программного обеспечения.

3.2 Системное программное обеспечение GCel PARIX.

PARIX - аббревиатура от PARallel unIX. Это системное программное обеспечение, построенное на базе стандартного программного обеспечения внешней машины (в нашем случае это Solaris 1.1) и дополненное программами, обеспечивающими функционирование GCel.

PARIX - распределенная программная система. Часть программ исполняется на внешней машине (SPARCstation-2), другая - в транспьютерных узлах.

Пользователь на этапе записи и редактирования текстов своих программ, компиляции и получения загрузочного модуля с GCel не работает. Он пользуется стандартной unix-средой на внешней машине (редакторы текстов, NFS и т.д.), специальными кросс-компиляторами, стандартными библиотеками, дополнительной специальной библиотекой с алгоритмами для параллельной обработки, а также специальным загрузчиком, который на внешней машине порождает загрузочный модуль для последующей загрузки транспьютерных узлов GCel.

Для ЛВС ВЦ РАН мы имеем 2 компилятора голландской фирмы ACE EXPERT - FORTRAN-77 и C ANSY. Компиляторы на выходе порождают объектный код в соответствии с промышленным стандартом COFF/T800.

Программа, созданная компилятором и загрузчиком, предполагает, что непосредственно на процессорных узлах будет исполняться не только программа пользователя, но также подпрограммы из т.н. USER RUNTIME LIBRARY, обращение к которым имели место из программы пользователя:

- программы обмена по линкам;
- другие программы, использующие модель программирования PARIX;
- стандартные численные подпрограммы.

Иными словами, для каждого транспьютерного узла на внешней машине подготавливается к исполнению программа в кодах транспьютера, содержащая коды алгоритма пользователя и коды тех программ из USER RUNTIME LIBRARY, которые используются данным алгоритмом.

Программа, исполняемая в процессорном узле, может запросить процедуру ввода-вывода (диск, экран, принтер). Эти процедуры реализуются через стандартный аппарат UNIXа - RPC, о котором уже подробно говорилось выше. Процессорный узел (транспьютер) выполняет роль клиента, а в качестве сервера исполняется специальный процесс на внешней машине - т.н. D-Server. На внешней машине может работать несколько D-Server'ов - максимально столько, сколько мы имеем USER ENTRY (в нашем случае - 3). Процесс "D-Server" на внешней машине образуется, когда пользователь выдает команду RUN. Внешняя машина от D-Server'a через управляющую сеть (в нашем случае, как говорилось выше, мы имеем один управляющий узел), передавая информацию через ENTRY USER 0 производит:

- инсталляцию пользовательской партии,
- загрузку исходного кода в каждый узел партии,
- инициализацию пользовательских процессов на каждом процессорном узле.

В дальнейшем D-Server вступает в работу, когда через механизм RPC ему передается заказ на ввод-вывод (внешний) или возникают аварийные ситуации (ошибки в программах, нарушение внешней среды и т.д.). D-Server при исполнении соответствующих функций может породить подчиненные процессы на внешней машине, которые в свою очередь могут быть процессами-клиентами по отношению к процессам-серверам на других машинах сети. Следует обратить внимание, что обмены по линкам между транспьютерными узлами реализуются непосредственно командами синхронных обменов транспьютера и D-Server не участвует в их реализации. Это способствует повышению общей эффективности решения задачи. То есть транспьютеры между собой обмениваются не средствами UNIX'a, а непосредственно с помощью команд обмена транспьютера T805. D-Server в режиме RPC "разговаривает" не с самими процессами задачи, а с ядрами (kernel) PARIX'a, которые загружены в каждый процессорный узел.

Следует отметить, что архитектуры PARSYTEC и PARIX появились в реальной практике пользователей, когда уже был некоторый опыт использования транспьютеров INMOS в других архитектурах, например, вместе с обычными персональными компьютерами (PC). Для таких архитектур получили распространение свои модели программирования (Helios с языком спецификаций CDL, MEIKOS, MACH, LINDA, EXPRESS и др.). Были разработаны специальные языки программирования для параллельной обработки (например, OCCAM).

Фирма PARSYTEC на GCel в принципе имеет возможность для поддержки большинства из сред (environment) ранних архитектур. Явно поддерживается CDL, в отношении других сред требуется программная системная настройка. Однако фирма PARSYTEC в большей мере рекомендует использование своей среды - т.н. модель PARIX, которая предполагает программирование на языках высокого уровня (без OCCAM'a) для некоторой виртуальной среды с соблюдением определенных правил. Представители фирмы PARSYTEC утверждают, что модель PARIX станет стандартом при использовании параллельной обработки для процессов с локальной памятью.

Приведем краткое описание модели PARIX. Фактически это описание дает пользователю, разработчику программ, использующих PARSYTEC GCell 1/64, методику разработки своих параллельных алгоритмов. Следует иметь ввиду следующие особенности модели PARIX:

- разрабатываемый параллельный алгоритм будет исполняться на множестве транспьютеров, называемом партицией (здесь под термином транспьютер понимается не физический процессор, как элемент транспьютерного узла с 4-мя линками, а некоторый виртуальный транспьютер с практически неограниченным числом линков). Для GCell 1/64 может быть партиция из 16 транспьютеров;
- p16 или lowerleft, или lowerright, партиция из 32 транспьютеров - p32 или upperhalf, или lowerhalf, партиция из 64 транспьютеров - p64 или GCell, или all; в GCell 1/64 нельзя использовать партиции другой длины кроме (16, 32, 64);
- программист имеет возможность писать универсальные программы в смысле количества используемых транспьютеров (16, 32, 64), для этих целей есть стандартная процедура NPROC (), выдающая число используемых транспьютеров;
- каждый транспьютер партиции имеет свою собственную память (4Мб), реальный объем памяти в значительной мере зависит от числа используемых виртуальных линков;
- в память каждого транспьютера будет загружена одна и та же программа. Это, пожалуй, главная особенность модели PARIX. Есть различные способы заставить исполнять различные программы на процессорах партиции при таком способе загрузки кода. Для этих целей имеются специальные процедуры: NOMPROC (), дающая ответ в виде номера транспьютера (для партиции длины N - номера транспьютеров от 0 до N-1), на котором выполняется данная программа. Следовательно, можно, используя обычный условный оператор, реализовать работу разных ветвей программы на разных транспьютерах. С помощью NOMPROC (), а также процедуры EXECUTE, динамически загружающей предварительно скомпилированную программу, можно построить собственный план загрузки каждого узла;
- каждый виртуальный транспьютер может быть связан с любым количеством других виртуальных транспьютеров, но два транспьютера могут при этом иметь только один общий линк с двумя подканалами (ввод-вывод);
- при реализации собственной программы можно обмениваться массивами любой длины по линкам между транспьютерами различными способами (различными процедурами). Но при этом всегда, как и для физического транспьютера, существует правило синхронного обмена: процедура ввода на одном транспьютере соответствует процедуре вывода на связанном с ним и соответственно наоборот.

Способы обменов следующие:

- синхронные обмены по виртуальным линкам (Send, Recv, SendLink, RecvLink, Select);
- асинхронные (точнее на GCell 1/64 псевдоасинхронные) обмены по виртуальным линкам (AInit, ASend, ARecv, ASync);
- синхронные случайные обмены между транспьютерами (SendNode и RecieveNode);
- для оптимальной работы транспьютеров с точки зрения обменов можно построить т.н. топологию (линия, круг, куб, гиперкуб, дерево и т.д.). Построить топологию можно с помощью стандартных процедур (MakeTorus, MakeTree и др.) и с помощью процедур построения топологии (newtop, addnewlink, freetop и др.) Последние позволяют динамически перестраивать и менять топологию.

- можно запустить параллелизм (подчиненные параллельные ветви) на одном транспьютере (только в "С", но не в FORTRANe);
- для реализации параллельных алгоритмов используются обычные с точки зрения входного языка стандартные компиляторы С ANSY и FORTRAN-77, никаких дополнительных "параллельных" операторов они не содержат. Весь параллелизм достигается использованием специальных "параллельных" процедур, о некоторых таких процедурах уже говорилось выше (nomproc, nproc, execute, send, recv и т.д.).

На наш взгляд модель PARIX для программиста достаточно естественна. Программист имеет один общий листинг программы, что облегчает отладку. И необходимо лишь привыкнуть к некоторой специфике программистского мышления, которая заключается в том, что отдельные части программы, ветви условного оператора, например, могут исполняться не на одном процессоре, а на разных. Особо следует продумать стратегию ввода-вывода исходных данных и конечного результата. При этом надо помнить, что внешний ввод-вывод связан с работой D-Server'a, аппарата RPC и поэтому протекает довольно медленно (секунды или даже десятки секунд против миллисекунд для внутренних обменов между транспьютерами).

Часто исходные данные для параллельной программы находятся во внешнем файле. В этом случае его надо прочитать один раз на один из транспьютеров партии, а дальше уже командами обменов разослать эти данные в полном объеме или частично по отдельным транспьютерам. Процедуру вывода данных также надо тщательно продумать. Вывод можно производить либо с каждого транспьютера партии, либо собрать все выходные данные на одном из транспьютеров партии, а затем, сформировав выходной файл, передать его во внешнюю память.

4. Принципы построения ИВС.

4.1 Особенности эволюции ИВС в 70 - 80 годах (обзор).

При создании современных информационно-вычислительных систем сегодня уже невозможно игнорировать опыт, который был накоплен мировым сообществом. Этого нельзя сделать ни по экономическим соображениям, ни вследствие того, что сама возможность развития технологий на современном уровне радикально зависит от всей совокупности уже достигнутого. Физически невозможно решить задачу построения современной высокоэффективной среды без освоения уже существующих информационных технологий и интеграции в мировое информационное сообщество. Этот опыт формировался на протяжении десятилетий в условиях, когда все решения проверялись временем, в условиях свободной конкуренции, когда концепции, казавшиеся идеальными сегодня, объективный процесс развития объявлял назавтра несостоятельными.

В данной главе мы попытаемся проанализировать в ретроспективе этот процесс и обосновать подходы, которые были использованы при разработке архитектуры системы, существующей в настоящее время в Вычислительном центре Российской академии наук.

Развитие вычислительной техники за прошедшие годы характеризуется прежде всего изменением структуры парка технических средств. В конце семидесятых годов основу парка составляли миниЭВМ, супермини-ЭВМ, большие универсальные машины (main-frame) и супер-ЭВМ. Прогресс микроэлектронных технологий и создание микропроцессора в значительной степени определили направление, по которому в восьмидесятых годах происходило развитие мини и супермини-ЭВМ. Именно успехи микроэлектроники позволили обеспечить снижение стоимости аппаратуры до такой

степени, что к началу восьмидесятых экономически оправданным стало создание персональных ЭВМ.

Персональные ЭВМ появились как результат эволюции мини-машин при переходе с элементной базы, использующей интегральные схемы малой и средней степени интеграции к большим интегральным схемам. Этот переход от мини-машин, которые в значительной части с самого начала были машинами почти индивидуального пользования, к действительно персональным машинам, оказался совершенно естественным и первоначально рассматривался многими только как просто переход к новой элементной базе. Характерно, что большинство производителей рассматривало эти машины просто, как новый вид мини-ЭВМ. Однако чрезвычайно низкая стоимость этих машин скоро определила их как новый сектор в вычислительной технике. Персональные машины начали завоевывать все новых и новых пользователей и скоро стало очевидным, что у них появилась собственная ниша, которая образовалась за счет привлечения в информатику людей, ранее никак не связанных (не профессионалов) с процессом обработки информации.

Этот сектор технических средств стимулировал возникновение и абсолютно нового рынка программных средств, ориентированных на нового пользователя. Это, прежде всего, "дружественные пользовательские интерфейсы", проблемно-ориентированные среды и инструментальные системы для автоматизации разработки прикладных программ.

Вместе с тем, мини-ЭВМ семидесятых годов явилась прародителем и другой ветви сегодняшних ЭВМ - 32-х разрядных машин, которые стали называть суперминиЭВМ. Этот переход оказался качественным скачком и позволил радикально увеличить производительность этого класса машин. Основной особенностью этих машин явилось наличие 32-х разрядного адреса и адресного пространства, соответствующего этому адресу. Именно наличие такого адреса (а не длина машинного слова - последние модели 16-ти разрядных машин могли работать с операндами длиной в 32 и 64 разряда) позволило устранить главное препятствие для увеличения производительности, поскольку обеспечило возможность прямого доступа к адресному пространству размером в 4 Гбайта, что практически сняло ограничение на объем оперативной памяти, предоставляемой каждой задаче, решаемой в ЭВМ.

Супермини-ЭВМ оказались пригодными для решения задач моделирования в системах проектирования, обработки и синтеза изображений и многих других, которые до этого решались только на больших машинах. Пионером в разработке и освоении выпуска машин класса супермини стала фирма Digital Equipment Corporation, которая в 1978 г. начала выпуск теперь широко известной ЭВМ VAX-11. Высокая производительность, гибкость, простота доступа к данным и относительно невысокая стоимость сделали 32-х разрядные машины основным средством проектировщиков, исследователей, позднее экономистов и большинства других профессиональных пользователей ЭВМ. Создание 32-х разрядного микропроцессора и разработка микросхем памяти емкостью более 1 Мбит привело к окончательному оформлению настольных систем высокой производительности, которые сегодня известны, как рабочие станции.

Ориентация рабочих станций на профессиональных пользователей в отличие от персональных ЭВМ ведет к тому, что рабочие станции - это хорошо сбалансированные вычислительные системы, в которых высокое быстродействие сочетается с большим объемом оперативной и внешней памяти, высокопроизводительными внутренними магистралями, высококачественной и быстродействующей графической подсистемой и разнообразными устройствами ввода-вывода. Это свойство рабочих станций выгодно отличает их от ПЭВМ и сегодня. Даже наиболее мощные ПЭВМ ряда IBM PC уже не в состоянии удовлетворить возрастающие потребности систем обработки из-за наличия в архитектуре этих машин ряда "узких мест". К тому же, стоимость рабочих станций непрерывно уменьшается и в ряде случаев уже не превышает стоимости персональных ЭВМ, превосходя их по своим характеристикам.

Большинство ведущих фирм выпускает свои рабочие станции. Среди фирм, активно действующих в этой отрасли, можно назвать такие как: APPLE, HEWLETT-PACKARD (HP), Digital Equipment Corporation, SUN. В результате предпринятых за последние годы усилий появился ряд новых быстродействующих архитектур и систем: параллельные системы, в том числе транспьютерные, системы на основе новых материалов (в том числе, арсенид галлия) и технологий. Среди этих новых разработок ЭВМ с сокращенным набором команд - RISC (Reduced Instruction Set Computer). Такие фирмы, как IBM, INTEL, MIPS, MOTOROLA, SUN выпустили свои RISC системы. Очень хорошие технические характеристики имеют рабочие станции серии RS/6000 корпорации IBM. Широко применяется известный и в нашей стране RISC процессор i860.

Существенное влияние на использование RISC систем оказали рабочие станции с архитектурой SPARC (Scalable Processor ARChitecture) фирмы SUN Microsystems. Значительный интерес представляет и последняя разработка фирмы Digital - 64 разрядный 200 Мгц RISC процессор ALPHA. При этом, однако, необходимо отметить, что сами по себе технические характеристики рабочих станций еще не определяют успеха изделий у конечного потребителя. Так, несмотря на прекрасные характеристики и широкую рекламу, рабочая станция RS/6000 не получила достаточно широкого распространения и заметно уступает многим другим (объемы поставок рабочих станций: SUN - 35,4%, Digital - 17,3%, HP/Apollo - 16,64%, Integraph - 5,1%, IBM - 3,4% (данные фирмы Dataquest, январь 1991 г.).

IBM, как известно, является одним из самых агрессивных производителей вычислительной техники в мире, и, тем не менее, ее усилия в этом секторе вычислительных средств до сих пор не принесли заметных результатов. Это положение определяется, по-видимому, тем, что IBM всегда стремилась создавать комплексные системы, рассчитанные на использование только собственного программного продукта, работающего только в программной среде IBM или ее партнеров. Вместе с тем, объем уже существующего прикладного обеспечения для рабочих станций, созданного помимо IBM, очень велик и игнорировать его наличие просто невозможно.

Разработчики аппаратуры и системного программного обеспечения не могут не учитывать этого обстоятельства. С другой стороны, фирмы - производители прикладных программных систем вынуждены учитывать ситуацию, когда завтра может появиться более мощная или более удобная станция, а значит, должна существовать возможность переноса программ с одной станции на другую. Сегодня уже невозможна ситуация, когда один мощный производитель диктует условия, создавая монопольный стандарт "де-факто". Характеристики выпускаемых изделий сейчас очень близки и совершенно невозможно предугадать, чья система завтра окажется лучше.

Таким образом, имеет место тенденция к унификации определенного слоя программной среды, которая должна обеспечить мобильность программ не только в рамках ряда изделий одного производителя, но и для различных ЭВМ, выпускаемых разными фирмами и сильно отличающихся друг от друга.

Существенным фактором, стимулирующим стандартизацию операционной среды является и широкое применение вычислительных сетей. Локальные, а тем более глобальные сети объединяют, как правило, совершенно разнородные вычислительные системы и мобильность программного продукта при работе в сетях играет чрезвычайно важную роль. К тому же, при работе в вычислительных сетях должна быть обеспечена и совместимость при доступе к данным (представление данных, доступ к базам данных с различными интерфейсами и на разных серверах).

Одним из наиболее важных факторов, определяющих сегодняшние тенденции в развитии информационных технологий является то, что фирмы, производящие вычислительную технику, ориентируются на рынок прикладных программных средств. Законодателями моды стали software'ные

фирмы, предоставляющие пользователям эффективные проблемно-ориентированные прикладные среды. Ведь в конечном итоге пользователю абсолютно безразлично, какого типа компьютер он использует, он выбирает не платформы как таковые, а программное обеспечение, которое позволяет решить его задачи.

Более того, поскольку программное обеспечение и информация стоят значительно дороже, чем аппаратные средства, пользователь всегда более консервативен, если речь заходит о смене прикладных программных средств и баз данных, нежели о смене аппаратной платформы. Производители аппаратных средств, ориентируясь на интересы пользователя, вынуждены это учитывать, если рассчитывают на коммерческий успех.

С другой стороны, как производители программного обеспечения, так и разработчики аппаратуры заинтересованы в постоянном улучшении характеристик своих изделий. А это возможно лишь при условии, что производители аппаратуры имеют возможность применять новые решения, что неизбежно влияет на существующие архитектуры и порой требует их радикальной замены. И в этом заинтересованы как первые, так и вторые. Следовательно, новые аппаратные решения также оказывают влияние на разработчиков прикладных систем, которые время от времени вынуждены обращать внимание на новые платформы. На сегодняшний день можно с определенностью утверждать, что наибольшего успеха может добиться лишь тот производитель, который в одинаковой степени поддерживает как разработку высокопроизводительных систем, так и обеспечивает удовлетворение наибольшего числа стандартов на мобильные программные средства.

К началу 90-х гг. уже явно обозначился отход от доминировавшей ранее технологии, основанной на использовании "главной машины" (mainframe), и переход к технологии "клиент-сервер", который по видимому стал поворотным пунктом в процессе развития информационных технологий. Этот переход окончательно разрушил монополизм нескольких гигантов в индустрии вычислительной техники, диктат которых к середине 80-х годов уже явно сдерживал реализацию новых тенденций в области обработки информации. Архитектуры, использующие технологию "клиент-сервер", основаны на организации распределенных вычислений с использованием многомашинных комплексов, связанных высокоскоростной сетью.

Технология "клиент-сервер" основана на организации распределенных вычислений на многомашинных комплексах, связанных высокоскоростной сетью. Первоначально в системах такого типа использовали распределение в моноархитектурной (однородной) сети. Примером такой системы является известная система VAXcluster, объединяющая до 32 ЭВМ типа VAX, фирмы Digital Equipment Corp. Характерной особенностью систем VAXcluster было использование специальных фирменных средств межмашинной связи, исключавших применение ЭВМ, выпускаемых другими фирмами. Тем не менее системы такого типа можно считать предшественниками современных систем, использующих технологию "клиент-сервер".

Со временем представление о распределенных вычислениях менялось, и следующим этапом ее реализации стала однородная вычислительная сеть миниЭВМ и позднее персональных ЭВМ. В такой системе пользователь получал локальную вычислительную систему и доступ ко всем ее ресурсам. Разработка и реализация концепции "Взаимодействия Открытых Систем" позволила создать неоднородные сети, состоящие из ЭВМ различной архитектуры и производительности. В этом случае пользователю уже обеспечивался доступ практически к любому типу ресурсов, включая возможность выбора любой ЭВМ в системе (от персональной до суперЭВМ).

С переходом к неоднородным сетям изменилась и точка зрения на саму сеть. Из средства обмена информацией сеть превратилась в средство для интеграции отдельных ресурсов (связанных с конкретными ЭВМ) в мощную распределенную вычислительную систему. Каждый элемент такой системы уже существует не сам по себе, а является органичной частью общего комплекса. Пользователь уже в состоянии выбирать наиболее адекватное его задаче подмножество вычислителей (серверов) таким образом, чтобы любая часть задачи решалась именно на той машине, которая

подходит для этого наилучшим образом. Не менее важно, что в системах такого типа имеются возможность их постепенной модернизации за счет замены отдельных устаревших технических и программных средств более современными.

В такой среде должны обеспечиваться следующие условия: мобильность, т. е. переносимость информации (интероперабельность среды), гибкость, позволяющая изменять конфигурацию (количество, состав аппаратуры и программного обеспечения), мобильность программ (возможность использовать одни и те же программные системы на различных аппаратных платформах) и мобильность персонала, т. е. возможность применять одни и те же человекомашинные интерфейсы на всех системах, входящих в среду.

Если способы обеспечения мобильности данных разработаны достаточно давно и базируются на реализации Международной Эталонной модели ВОО (OSI/ISO - Open System Interconnection) или национальных стандартах (таких, как совокупность протоколов TCP/IP- Transmission Control Protocol/Internet Protocol), то до сравнительно недавнего времени единственным способом обеспечения мобильности программного обеспечения (ПО) была стандартизация архитектуры. Хорошо известна концепция IBM, реализованная в архитектурах серий 360 и 370 или в сериях ЭВМ PDP-11 и VAX-11 фирмы Digital Equipment. Однако любая, даже самая передовая для своего времени архитектура неизбежно устаревает и уступает место другим и вопрос о переносе программ на другие архитектуры все равно остается. (Попутно заметим, что задача мобильности программного обеспечения связана не только, и даже не столько, с системами типа "клиент-сервер", а является одной из самых старых и вполне самостоятельных проблем в программировании.)

Проблему мобильности программного обеспечения сегодня необходимо решать на основе его независимости от аппаратной платформы. Одной из таких основ уже в течение многих лет являются алгоритмические языки. Однако в течение долгого времени мобильность ПО, которая обеспечивалась алгоритмическими языками, не могла быть реализована в полной мере. На ЭВМ прошлых поколений с их низким быстродействием, оперативной памятью и адресным пространством небольших размеров невозможно обеспечить достаточной эффективности программ, создаваемых с помощью алгоритмических языков. Высокая удельная стоимость единичных ресурсов вынуждала разработчиков заботиться об их экономном использовании. В частности, характерной тенденцией было стремление получать как можно более короткий исполняемый код. Для этого ЭВМ применяли сложные, многотактовые команды, имевшие значительную функциональную избыточность (CISC-архитектура).

Положение резко изменилось лишь тогда, когда микроэлектронная технология перешла на субмикронные проектные нормы. С уменьшением размеров элементов значительно возросли тактовые частоты и количество элементов, интегрируемых в одном кристалле. Как следствие, резко повысилась быстродействие и снизилась стоимость всех основных компонентов ЭВМ. К концу 80-х годов стала экономически целесообразной установка оперативной памяти емкостью в несколько мегабайт даже на персональные ЭВМ, а быстродействие микропроцессоров возросло до десятков миллионов операций в секунду. Одновременно с этим возросла емкость внешних накопителей.

Эти изменения привели к снижению роли ресурсных ограничений на архитектуры ЭВМ. Производители обратили внимание на архитектуру RISC, предложенную в начале 80-х. Первый RISC-процессор был сделан в 1982 году. Это событие не вызвало в то время больших откликов, однако оно в значительной степени определило пути развития ИВС до конца десятилетия и играет решающую роль и сегодня. Во-первых, RISC архитектура обеспечила существенное повышение производительности микропроцессоров, а во-вторых предоставила, наконец, аппаратную базу для реализации эффективной переносимости программ для процессоров разных производителей.

RISC процессоры вне зависимости от конкретных реализаций, принадлежащих различным производителям, имеют ряд общих, характерных в совокупности именно для них особенностей:

- большинство машинных команд исполняется за один машинный такт;
- регистровая архитектура: все команды обработки выбирают операнды и помещают результаты только в регистрах. Обращение к ОЗУ производится только командами чтения/записи;
- конвейерная обработка команд, которая позволяет исполнять несколько команд одновременно;
- простой фиксированный формат команд с небольшим количеством методов адресации ОЗУ;
- наличие быстродействующей памяти: любая RISC машина имеет по крайней мере 32 регистра общего назначения и большую кэш-память;
- упрощенный набор команд, из которого исключены многоцикловые команды, предназначенные для реализации сложных функций, что обеспечивает предельно короткий машинный цикл и высокое быстродействие аппаратуры.

Эти особенности приводят к тому, что сложные многоступенчатые функции перемещаются в область программной реализации. В результате ручное программирование становится малоэффективным, и основную роль начинают играть именно языки высокого уровня. Исполняемый машинный код становится длиннее и требует ОЗУ большего объема, чего так стремились избежать разработчики традиционных CISC архитектур.

Характерная для архитектуры RISC элементарность набора команд позволяет приблизить эффективность программ, написанных на языках высокого уровня, к эффективности программ в машинном коде и автоматизировать процесс настройки программ для их оптимизации. В результате использование стандартных компиляторов, сделало возможным обеспечить на уровне языков высокого уровня эффективную мобильность программ.

Однако, мобильность программ, как таковая, является важным, но не единственным аспектом в решении задач мобильности программного обеспечения в целом. Исключительно важным аспектом является обеспечение мобильности системной платформы. В традиционном подходе эта задача может рассматриваться как создание мобильной операционной системы (ОС). Такая система была разработана достаточно давно – это ОС UNIX.

Хотя ОС UNIX была разработана еще до появления персональных ЭВМ, она не могла эффективно использоваться, так как требовала значительных аппаратных ресурсов. С появлением мощных RISC-микропроцессоров с 32-х разрядной архитектурой UNIX проявила себя как наиболее перспективная мобильная операционная среда. Исторически ОС UNIX оказалась самым жизненным вариантом для создания общей базы переносимости.

Одна из причин, по которой система UNIX рассматривается как базовая мобильная ОС, состоит в том, что она почти целиком написана на языке высокого уровня C, модульна и относительно гибка. UNIX состоит из основных компонентов, включающих ядро, инструментальные утилиты и оболочку. Ядро, составляющее сердцевину UNIX`а, состоит из относительно небольшого набора программ, предоставляющих системные ресурсы и непосредственно взаимодействующих с аппаратурой.

Прикладные программы, создаваемые для работы в UNIX, при определенных условиях могут иметь весьма высокую переносимость как в другие UNIX-подобные системы, так, во многих случаях, и в системы, удовлетворяющие стандартам на интерфейсы, подобные тем, которые разработаны различными организациями, такими как X/Open и POSIX. Вместе с тем, хотя ОС UNIX машинно-независима, программы, которые реализуют некоторые службы и часть кода, зависят от аппаратуры. Деятельность таких организаций, как UniForum, POSIX и X/Open, направлена на поиск общего функционального ядра, которое позволило бы достичь переносимости между различными системами. В стадии исследований и разработки

сегодня находится ряд систем, специально проектируемых, исходя из требований, предъявляемых разнородной распределенной сетевой средой. Некоторые из них могут со временем стать хорошей мобильной системной платформой.

Унификация интерфейсов, протоколов взаимодействия и способов представления данных в информационно-вычислительной среде является исключительно важной задачей, которая, однако, не может быть решена однократно единственным образом. Возможности ИВС постоянно возрастают, появляются новые технологии, новые типы и способы обработки и передачи данных, более мощные аппаратные платформы, которые активно влияют на организацию среды. В условиях жесткой конкуренции между различными производителями вычислительной техники и программного обеспечения создание системы глобального масштаба является исключительно сложной задачей, которую невозможно решить только административными методами. Этот процесс протекает в течение длительного времени, интеграция осуществляется путем достижения соглашения между всеми участниками этого процесса и принятия объективно сложившихся в индустрии решений (стандарты de-facto) в качестве международных стандартов.

В этих условиях сформировалась концепция обобщенной программной среды, получившая название Открытые Системы (Open Systems). Открытые системы - это совокупность стандартов на различные компоненты операционной среды, предназначенных для обеспечения мобильности программного продукта в рамках неоднородной, распределенной вычислительной системы.

4.2 Функциональная структура программно-аппаратной среды современных информационно-вычислительных систем.

Существующие стандарты относятся к способам и методам взаимодействия компонент как программных, так и аппаратных средств, составляющих системную и прикладную среду ИВС.

Одной из наиболее сложных задач в этом смысле является определение структуры такой среды, т.е. описание ее в виде совокупности определенных функциональных элементов. Именно такое описание позволяет обеспечить упорядоченную стандартизацию всех возможных вариантов взаимодействия. На сегодняшний день существует множество подходов к разбиению среды. Процесс стандартизации среды неизбежно связан с разрешением противоречий между подходами различных производителей к такому разбиению. Речь идет о создании модели среды. Наличие множества таких моделей иллюстрирует разнообразие таких подходов.

Вместе с тем за последние годы сделан ряд решительных шагов в выработке общих подходов к решению этой задачи. Рядом международных и национальных организаций предложены собственные варианты моделей среды. Одним из таких вариантов является модель OSE (Open System Environment), предложенная комитетом IEEE POSIX.

Национальным институтом стандартов и технологий США на основе OSE POSIX был выпущен документ "APPLICATION PORTABILITY PROFILE APP. The U.S. Government's Open System Environment Profile OSE/1 Version 2.0". Этот документ определяет рекомендуемые для федеральных учреждений США спецификации в области информационных технологий, обеспечивающие мобильность персонала, системных и прикладных программных средств. Этому документу следуют все ведущие производители вычислительной техники и программного обеспечения в США.

APP строится как профиль открытой среды, предназначенный для использования федеральными ведомствами. Он охватывает широкую область прикладных систем, представляющих интерес для многих федеральных организаций, однако не включает всего списка используемых прикладных систем. Индивидуальные стандарты и спецификации, входящие в APP, определяют форматы данных, интерфейсы, протоколы и/или их комбинации.

Рассмотрим структуру среды OSE, используемую в APP.

4.2.1 Функциональные области APP.

Все виды функционального обслуживания в рамках APP могут быть представлены следующими семью функциональными областями:

- а) функции, реализуемые операционной системой;
- б) функции, реализующие человеко/машинные интерфейсы;
- в) управление данными;
- г) обмен данными;
- д) поддержка разработки программного обеспечения;
- е) машинная графика;
- ж) сетевые функции.

4.2.2 Область функций операционной системы.

Функции, реализуемые операционной системой, являются важнейшими функциями, обеспечивающими управление прикладной платформой. Эти функции обеспечивают интерфейсы для взаимодействия прикладных программ и платформы. Область этих функций включает следующие операции:

- а) Функции ядра операционной системы – являются функциями нижнего уровня и применяются для создания и управления процессами исполнения программ, генерации и передачи сигналов операционной системы, генерации и обработки сигналов системного времени, управления файловой системой и каталогами, управления и обработкой запросов ввода/вывода и обслуживанием внешних устройств.
- б) Команды и утилиты – механизмы для исполнения функций уровня оператора, такие как сравнение, печать и отображение содержимого файлов, редактирование файлов; поиск образцов; регистрация сообщений; перемещение файлов из каталога в каталог; сортировка данных; исполнение командных строк и доступ к служебной информации системы.
- в) Расширения реального времени – функции, реализующие прикладные и системные интерфейсы, которые используются в прикладных областях, требующих детерминированного исполнения, обработки и реакции. Расширения этого типа определяют прикладные интерфейсы к базовым функциям операционной системы: ввода/вывода, доступа к файловой системе и управления процессами.
- г) Управление системой – функции, позволяющие создавать и управлять ресурсами, предоставляемыми пользователю, правами доступа к подсистемам и устройствам, управление конфигурацией и производительностью устройств, файловой системы, административными процессами, авторизацией доступа, поддержкой живучести системы.

4.2.3 Человеко-машинные интерфейсы.

Человеко-машинные интерфейсы определяют методы, с помощью которых пользователи могут общаться с прикладными системами. В зависимости от различных условий, которые могут определяться как пользователями, так и прикладными системами, интерфейсы этого типа могут обеспечивать следующие функции:

- а) Операции типа "клиент-сервер" – определяют взаимоотношения между процессом "клиентом" и процессом "сервером" в сети, в

частности, между процессами, имеющими место при отображении с помощью графического пользовательского интерфейса. В этом случае программа, которая управляет каждым дисплейным устройством, реализует процесс-сервер, в то время как пользовательская программа представляет процесс-клиент, который запрашивает обслуживания сервером.

- б) Определение объектов и управление – включает спецификации, с помощью которых задаются характеристики отображаемых элементов: цвет, форма, размеры, движение, графические характеристики, взаимодействие между отдельными элементами и т.д.
- в) Параметры окон – спецификации, которые позволяют определить как создаются окна, передвигаются, сохраняются, восстанавливаются, удаляются и взаимодействуют друг с другом.
- г) Поддержка диалога – спецификации, с помощью которых устанавливаются взаимоотношения между тем, что отображено на экране (в т.ч. движение курсора, данные, введенные с клавиатуры и дополнительных устройств), и тем, как меняется изображение в зависимости от вводимых данных.
- д) Спецификации мультимедиа, включая API – спецификации, определения функций и форматов данных, которые поддерживают манипуляции различными формами цифровой и аналоговой аудиовизуальной информации в рамках единой прикладной системы.

Пользовательские интерфейсы являются едва ли не самой сложной областью в разработке и эксплуатации. В течение последних нескольких лет в области технологии пользовательских интерфейсов получены заметные результаты как в части связанной с пользователями, так и в области создания инструментальных средств для построения систем.

4.2.4 Функциональная область поддержки разработки программного обеспечения.

Функциональная область поддержки разработки программного обеспечения предоставляет для этого необходимую инфраструктуру, в которую входят как языки программирования, так и интегрированные инструментальные системы. В этой функциональной области можно выделить следующие средства:

- а) Стандартные языки программирования вместе со средой поддержки, в которую входят отладчики, средства настройки и оптимизации кода, редакторы.
- б) Интегрированные среды и инструментальные системы для разработки программного обеспечения, включающие системы и программы для автоматизированного создания и поддержки программного обеспечения. В состав таких систем входят средства для выбора спецификаций и анализа прикладных систем на этапе проектирования, для создания и тестирования программного кода, документации и средств поддержки коллективных проектов для групп разработчиков. Интерфейсы, входящие в состав этих средств, обеспечивают хранение, выборку и обмен информацией между программами среды.

4.2.5 Функциональная область управления данными.

Центральной задачей для значительного большинства систем является управление данными. Функции этой области могут определяться независимо от процессов, порождающих и использующих данные. Функции этого типа обеспечивают возможность независимой манипуляции данными, совместное или независимое их использование различными процессами. Системы управления данными реализуют следующие функции:

- а) Обслуживание доступа к словарям и каталогам данных, которые обеспечивают программистам и пользователям доступ к информа-

ции о данных (мета-данным). Мета-данные могут включать внутренние и внешние форматы, правила, обеспечивающие сохранность и секретность, и могут располагаться в распределенных системах.

- б) Системы управления базами данных (СУБД) обеспечивают управление доступом к структурированным данным. СУБД позволяют обращаться к комбинациям данных, расположенным в различных схемах. Базы данных доступны через интерфейсы, входящие в стандарты языков программирования или интерактивные интерфейсы языков четвертого поколения. СУБД также обычно реализуют особые функции создания, размножения, перемещения, резервного копирования/восстановления и архивации баз данных, хотя некоторые из них предоставляются также и файловой системой, относящейся к функциональной области операционных систем.
- в) Функции распределенного доступа обеспечивают обращение к данным, хранящимся в удаленных базах.

4.2.6 Функциональная область обмена данными.

Функциональная область обмена данными обеспечивает поддержку специфических аспектов обмена информацией, включая форматы данных и их семантику, как для программ, работающих на одной платформе, так и на различных (неоднородных) платформах. Эта область включает спецификации для функций обмена следующих типов:

- а) Документы - это спецификации для кодирования данных (текст, рисунки, числа, специальные символы и т.д.) и как логические, так и визуальные структуры электронных документов.
- б) Графические данные - независимые от устройств определения элементов рисунков.
- в) Производственные данные - спецификации, описывающие технические рисунки, документацию, данные другого типа, необходимые в производстве и проектировании продукции, включая геометрические или не геометрические данные, характеристики форм, допуски, свойства материалов и т. п.

Существуют различные уровни сложности представления данных, используемые в процессе обмена. Самый простой, нижний уровень сложности (уровень 1), обеспечивает возможность задать представление данных, участвующих в обмене. Представление данных может определяться явным образом, путем указания формата, либо путем ссылки на язык программирования. Следующий, уровень 2, отражает содержание одиночного объекта. Примерами спецификаций такого типа могут быть тексты, растровые изображения или аудиоинформация. Уровень 3 включает спецификации для представления сложных объектов, состоящих из элементарных объектов, соответствующих уровню 2. Уровень 4 - это уровень языка представления данных. Последний, пятый уровень - уровень приложений, который может использовать любые из нижних уровней для обмена с другими прикладными программами.

4.2.7 Область графических функций.

Эта функциональная область предоставляет функции, используемые для создания и манипуляций с отображаемыми изображениями. Функции такого рода обеспечивают определение и поддержку отображаемого элемента и определение атрибутов изображения. Функции этой области определены в спецификациях многомерных графических объектов и изображений и в форме, независимой от конкретных устройств. Доступность и целостность как функций, поддерживающих разработку программного обеспечения для изображений и графики, а также самих графических данных обеспечивается за счет средств, обеспечивающих безопасность для данной области.

4.2.8 Функциональная область сетевой поддержки.

Сетевые функции предоставляют механизмы для поддержки прикладных систем, которым требуется интероперабельный доступ к данным и программам в неоднородной сетевой среде. В этой области имеются следующие механизмы:

- а) Собственно коммуникации, которые включают API, и спецификации протоколов для прозрачной и надежной передачи данных через коммуникационные сети.
- б) Прозрачный доступ к файлам, расположенным в любом месте неоднородной сети.
- в) Поддержка персональных ЭВМ и микро-ЭВМ для обеспечения интероперабельности с системами, базирующимися на различных операционных системах (ОС), в частности микрокомпьютерных ОС, которые могут оказаться не соответствующими международным и национальным стандартам.
- г) Дистанционное обращение к процедурам - спецификации для обращения к процедурам, расположенным во внешней распределенной среде.

4.2.9 Интегрально поддерживаемые функциональные области.

К этой области относятся функции, которые интегрируются внутри уже описанных областей, и которые в рамках данной референсной модели затруднительно выделить в самостоятельные элементы, поскольку эти функции в каждом конкретном случае имманентно связаны с функциями каждой из рассмотренных функциональных областей.

К этим функциям относятся функции, обеспечивающие защиту в компьютерной среде и функции системной поддержки и администрации.

5. Поддержка, сопровождение и администрирование в ИВС ВЦ РАН.

5.1. Общие вопросы.

На стадии проектирования ИВС ВЦ РАН мы полагали, что поддержка работоспособности программно-аппаратного комплекса должна решаться на качественно новом современном уровне. Это подсказывал нам личный опыт, накопленный в работах по поддержке и сопровождению больших систем коллективного пользования.

К идее качественно и целенаправленно организовывать работы по поддержке программно-аппаратных комплексов пришла и наука о программировании. Широкое распространение открытых вычислительных систем и операционной системы UNIX во многом были предопределены смыканием реальных запросов практики на новые принципы организации программно-аппаратных комплексов и большой теоретической исследовательской работой, проделанной в этой области.

В середине 80-х годов разразился т.н. "кризис программирования", когда при резко снизившейся стоимости аппаратных средств не удавалось понизить затраты на программирование. Более того, эти затраты возрастали с увеличением объема и сложности программного обеспечения, системного и прикладного. В этот период были тщательно исследованы вопросы методологии программирования на разных этапах жизненного цикла, активно развивались исследования по технологии программирования, по оценке качества программ и т.д.

В частности стало неожиданно ясно, что основные затраты средств в жизненном цикле программ приходятся на этап сопровождения, т.е. на этап модификации программ на стадии их использования. По некоторым оценкам эти затраты составляли 70–80%. Иными словами, самым дорогим оказался этап жизненного цикла, который часто полностью игнорировался разработчиками. Как следствие, сопровождение было местным, что имело целый ряд недостатков и еще более увеличивало интегральную стоимость вычислительных систем. В то же время архитектура программных средств была такова, что настройку вычислительных систем на местную среду использования приходилось реализовывать через модификацию программ.

Унификация аппаратных средств, стандартизация в области системного программного обеспечения, авторское сопровождение программ, развитие разработчиком специальных программных инструментальных средств, развитие средств автоматизации на этапе сопровождения – вот те основные мероприятия, которые во многом способствовали выходу из "кризиса программирования".

Идеология открытых вычислительных систем на базе стандартов UNIX'a, стандартизация в области систем программирования и прикладных пакетов, принятые нами за основу при использовании ИВС ВЦ РАН, отвечают самым современным взглядам на вопросы поддержки системного программного обеспечения и нашему пониманию этого вопроса.

Основные положения организации поддержки программного обеспечения для программного комплекса ИВС ВЦ РАН следующие:

- местное сопровождение должно сводиться практически к нулю, модификация программ (на уровне текстов и кодов) должна отсутствовать;
- исправление ошибок и развитие программного обеспечения должны осуществляться разработчиком, который обязан регулярно выпускать новые версии компонент системного программного обеспечения;
- должны использоваться только стандартные аппаратные средства, приспособленные для работы в открытой среде;
- создание местной среды должно осуществляться через параметрическую настройку компонент программного обеспечения и модификацию специально предназначенных для этой цели командных текстовых файлов (скрипт-файлов на языках SHELL и CSHELL).

В развертываемой в ВЦ РАН вычислительной среде эти принципы реализованы в полной мере. Поддержка программного обеспечения реализуется через службу администрирования по определенной методике на базе специально разработанных средств. Администратор системы становится обязательной ключевой фигурой с вполне определенными обязанностями.

Главные задачи службы администрирования:

- правильно установить и настроить отдельные компоненты системы (как аппаратные, так и программные), обеспечить взаимодействие этих компонент;
- распределять ресурсы системы (системные и пользовательские);
- обеспечить соответствующими средствами защиту программных ресурсов и надежность сохранения информации;
- осуществлять регулярный контроль за использованием ресурсов и оптимизацию их использования;
- отслеживать состояние работы системы в целом и ее отдельных компонент, производить соответствующие системные измерения, с помощью которых выявлять "узкие" места в системе, производить реконфигурирование системы;
- осуществлять поиск и фиксирование ошибок в работе аппаратно-программных средств и организовывать их устранение.

Местное сопровождение, т.е. модификация текстов и кодов программ, практически отсутствует, как было сказано выше.

Отчасти к местному сопровождению можно отнести обязанность административной службы пополнять программное обеспечение за счет установки новых программных компонент и устанавливать новые версии про-

граммного обеспечения. То есть модификация программного обеспечения производится только за счет его расширения и замены отдельных блоков на другие блоки более поздних версий.

В процессе работы складывается определенная методика административной работы:

- сбор информации о новых разработках, новых версиях существующего программно-аппаратного обеспечения системы;
- сбор информации от пользователей по вопросам качества работы системных компонент, улучшения комфортности среды, расширения функций среды;
- проведение административной работы и местного сопровождения с тщательным фиксированием проделанной работы в специальном журнале административной службы;
- связь со службами центрального сопровождения в соответствующих фирмах;
- постоянное совершенствование собственных знаний, организация
- повышения квалификации пользователей.

Ниже мы рассмотрим конкретную работу службы администрирования на примере поддержки отдельных компонент среды ИВС ВЦ РАН. Таблицы и схемы по техническому и программному обеспечению даны в приложении в виде таблиц. В разделе 5 существенно используется техническая документация по ЭВМ Sun SPARCstation и ее программному обеспечению (Solaris 1.1).

5.2. Организация работы с внешней памятью в среде ИВС ВЦ РАН.

5.2.1. Аппаратные средства внешней памяти и их настройка.

Организация работы с внешней памятью является базисом, во многом определяющим функционирование вычислительной системы в целом, ее эффективность и надежность.

Аппаратные средства среды ИВС ВЦ РАН имеют разнообразную внешнюю память:

- магнитные диски и массивы магнитных дисков;
- CD-ROM (оптические диски, работающие только на чтение);
- накопители на магнитной ленте.

На физическом уровне все устройства внешней памяти являются периферийными устройствами рабочих станций. Все периферийные устройства внешней памяти взаимодействуют с центральной частью через адаптер SCSI (Small Computer System Interface). SCSI – это промышленный стандарт, шина, обеспечивающая высокоскоростной параллельный обмен данными. На каждой рабочей станции может быть до 8 SCSI-устройств (номера 0-7). Все 8 устройств включаются последовательно и по умолчанию имеют следующие номера:

- 0, 1, 2, 3 – диски
- 4, 5 – магнитные ленты
- 6 – CD ROM
- 7 – хост-ЭВМ

Оптические диски (CD-ROM) используются главным образом для установки программного обеспечения. Все программные продукты поставляются, как правило, на таких дисках.

Устройства на магнитной ленте в основном используются как инструмент для копирования (backup), хотя иногда на магнитной ленте поставляются некоторые программные продукты.

Магнитные диски являются основными устройствами для создания распределенной файловой системы.

Файловую систему, построенную на существующих физических ресурсах, активно используют пользователи и многие системные программы.

Посмотрим, как реализована в операционной системе рабочих станций (SunOS) распределенная файловая система и какие имеются средства ее поддержки и администрирования.

На физических ресурсах рабочих станций может быть построена распределенная файловая система (NFS). Это осуществляется следующей последовательностью действий:

- форматирование всех дисковых устройств с разбиением их на логические устройства (партиции);
- создание файловых систем в партициях;
- заполнение таблиц для монтирования;
- запуск на рабочих станциях процессов, управляющих монтированием и работой файловых систем;
- монтирование файловых систем.

Тут будет уместно сказать о множественном толковании термина "файловая система" в UNIX'e. Под этим термином понимают и всю совокупность файлов в вычислительном комплексе, и совокупность файлов на отдельной рабочей станции, и множество файлов на отдельном логическом устройстве.

Рассмотрим этапы создания распределенной файловой системы более подробно.

Форматирование диска - это его своеобразная разметка. В результате форматирования создается т.н. "сырой" диск (raw-disk), разбитый на логические устройства - партиции. В результате форматирования каждая поверхность диска будет разбита на дорожки (треки), сумма дорожек для одного положения головок образует цилиндр. Дорожки разбиты на блоки (секторы) длиной 512 информационных байт (есть диски с секторами большей длины). Каждый сектор имеет заголовок (header) для идентификации сектора и "хвост" (trailer), где хранится контрольная сумма сектора. С дорожками, секторами и цилиндрами с дисковой памятью на нижнем уровне работает программный драйвер, который запускает физические команды через устройство управления диском (контроллер).

Форматирование осуществляется специальной командой FORMAT, которая обязательно содержится в т.н. "малом" UNIX'e (MUNIX - Memory UNIX), целиком резидентно работающем в оперативной памяти. Кроме процедуры FORMAT в MUNIX'e имеется специальный файл FORMAT.DAT, содержащий всю информацию для различных типов диска. При исполнении команды FORMAT отыскивается нужная информация, соответствующая данному типу диска (тип контроллера, физические и геометрические параметры диска, количество цилиндров - основных и альтернативных и т.д.), а также информация о разбиении диска на партиции (таблица партиций). В результате работы команды FORMAT вся эта информация переносится на диск в специальную запись, называемую меткой (LABEL). Вообще говоря, фирменные диски поставляются, как правило, уже отформатированными, т.е. LABEL уже содержит всю необходимую информацию как о типе диска, так и о разбиении на партиции. Это, естественно, не исключает возможность переразметки таких дисков и задание новой таблицы партиций.

Логический диск или партиция могут иметь имена "a", "b", "c", "d", "e", "f", "g", "h". Все физические диски имеют стандартные идентификаторы, причем имя зависит от типа интерфейса. Например, для SCSI-дисков имена будут "sd0", "sd1", "sd2" и т.д. Как уже говорилось выше, диск, который отформатирован, но на котором еще не размещены файловые системы, называется "сырым" (raw). Имя raw-диска начинается с буквы "r". Например, "rsd0". Имя логического диска является составным. Например, "rsd1b" (в конце имени - имя партиции). Все имена "сырых" логических устройств в виде псевдофайлов присутствуют в каталоге "/dev".

Разбиение физического диска на логические (партиции), вообще говоря, произвольно. Но для унификации предлагаются некоторые канонические варианты разбиения. Например, для автономных рабочих станций (standalone) рекомендуется завести партиции "a", "b", "c", "g", и "h". Причем, в таблице партиций для партиции "c" показано все множество цилиндров, т.е. весь диск. Это рекомендация для таблиц партиций всех видов рабочих станций.

Следующий этап - это создание структур файловых систем. Для каждой партиции, которая будет содержать файловую систему, запускаются процедуры "newfs" и "mkfs".

После того, как выполнена команда "newfs", она передает соответствующую информацию команде "mkfs", которая в действительности и создает структуру файловой системы. При установке системы установочная процедура SunInstall через команды "newfs" и "mkfs" создает файловые структуры, а затем размещает конкретные файловые системы на определенных логических устройствах (партициях). Можно, конечно, заполнить файловую систему новым содержимым и на работающей рабочей станции обычным копированием.

В UNIX'e имеется несколько разновидностей файловых систем, отличающихся в деталях своими структурами. В нашем случае мы имеем дело с распределенной файловой системой NFS. Не останавливаясь подробно на структурах файловых систем, отметим лишь следующее. Файловая система хранит непосредственно на диске управляющие структуры:

- метки;
- суперблок;
- блоки групп цилиндров;
- таблицу "inodes" (index node) и непосредственно данные в виде групп цилиндров, каждый из которых имеет и "inodes" и блоки данных.

Структура файловой системы такова, чтобы с одной стороны обеспечить удобную работу с файлами, в частности, используя аппарат ссылок и монтирования, а с другой - надежность и сохранность информации как от несанкционированного доступа, так и при определенном виде сбоях системы.

Принято каноническое размещение файловых систем по партициям:

партиция "a" - корневая (root - "/") файловая система с поддиректориями "/etc", "/var" и др.

партиция "b" - здесь не создается файловая система, это область "подкачки" (swap), работа с которой производится ядром системы, средствами процедур нижнего уровня; через аппарат "swap" обеспечивается динамическая работа с оперативной памятью.

партиция "g" - "/usr" - в этой файловой системе находятся системные библиотеки, команды операционной системы, а также все программное обеспечение верхнего уровня (компиляторы, библиотеки, пакеты и т.п.).

партиция "h" - "/home" - это файловая система, содержащая дерево всех файловых систем пользователей.

Партиция "c", как уже говорилось, содержит весь физический диск. Партиции "d" и "e" заводятся только на файловом сервере (об этом ниже), в "d" выделяется дисковое пространство в виде "/export", в котором будут храниться корневые каталоги бездисковых клиентов, а в "e" - "/export/swap" - области "подкачки" (swap) этих клиентов.

Партиция "f" - произвольного использования.

После того, как файловые системы созданы и заполнены на логических устройствах, происходит этап монтирования, т.е. из отдельных файловых систем на логических устройствах создается общая единая распределенная файловая система (NFS).

К моменту, когда должно быть начато монтирование файловых систем, уже должно быть определено, какие рабочие станции являются файловыми серверами, а какие клиентами. В общем случае в сети может быть несколько серверов. Кроме того, одна и та же станция может быть сервером и клиентом одновременно. На рабочих станциях-серверах определяют, какие файловые системы на его физических ресурсах будут доступны через сеть для станций-клиентов. Про такие файловые системы говорят, что они будут экспортироваться для клиентов. Обычно экспортируются домашние директории (типа /home) с тем, чтобы был доступ к своим файлам у пользователей со всех рабочих станций. Многие элементы программного обеспечения такие как, библиотеки, компиляторы, графические системы и т.п. также размещаются на сервере, который экспортирует их клиентам (/usr).

Одна и та же файловая система может экспортироваться для всех клиентов, а может только для некоторых. Экспортировать можно в режиме чтения и записи (rw), а можно только чтения (ro). Имеется еще ряд параметров для экспортирования. Вся информация об экспортируемых файловых системах содержится в файле "/etc/exports" на серверах. Выполнив команду "exportfs", можно осуществить экспорт файловых систем. Если на данной рабочей станции файл "/etc/exports" существует, то на этапе начальной загрузки (boot) выполняются команды "/usr/etc/nfsd" и "/usr/etc/rpc.mountd", которые запустят два процесса-"демона" (daemon) на сервере.

Далее надо заполнить на всех станциях файл "/etc/fstab". В этом файле задается информация о монтируемых файловых системах (локальных и распределенных). При загрузке операционных систем на этих станциях локальные файловые системы будут смонтированы сразу же соответствующей командой "mount" в системном командном файле. Удаленные файловые системы будут монтироваться, используя механизм "клиент-сервер". Через RPC на сервер поступит запрос к демону "rpc.mountd", и этот демон вернет ссылку, называемую "file handle", на соответствующую директорию. Клиент, получив ответ, разместит "file handle" в соответствующей таблице "/etc/mtab".

Если все таблицы (exports, fstab) составлены правильно, то процесс монтирования завершается успешно - распределенная файловая система создана. Теперь любой клиент может работать с любой файловой системой сервера (смонтированной) так, как будто эта файловая система находится на клиенте. При обмене удаленными файлами через сеть функционируют демон "nfsd", запускаемый на сервере и демон "biod", запускаемый на клиенте. Для повышения эффективности обычно на сервере запускается до 8 копий "nfsd", а на клиентах - до 4-х копий "biod".

Этапом монтирования завершается процесс создания файловых систем.

После завершения этапа монтирования общая распределенная файловая система собрана. Отдельные дополнительные "веточки" файловой системы могут появляться динамически через специальный аппарат автомонтирования (automount). Файловые системы монтируются через автомонтировку только тогда, когда идет первое обращение к таким файловым системам.

5.2.2. Работа с гибкими и компакт-дисками.

На рабочих станциях сети, как уже говорилось выше, пользователям предоставляется возможность работать с компакт-дисками (CD-ROM) и дискетами (3 1/4"). Чтобы облегчить "монтирование" (mount) и "размонтирование" (umount) соответствующих устройств создан аппарат псевдопользователей - "mfd", "mcd", "umfd", "umcd". Пользователь вставляет дискету или диск в соответствующее устройство и в ответ на приглашение LOGIN задает псевдо-имя "mfd" или "mcd". В результате содержимое

дискеты или диска становится доступным через каталоги `"/pcfs"` и `"/cdrom"` соответственно.

Подключившись к системе под своим входным именем, пользователь может читать с компакт диска, с дискеты и писать на дискету. После окончания работы с этими устройствами и завершения сеанса, пользователь должен вновь в ответ на приглашение LOGIN набрать `"umfd"` или `"umcd"`, что позволит выполнить команду "размонтирования" и команду `"eject"`, освобождающую устройства от дискеты или диска. И, наконец, пользователь может работать с гибкими дисками на персональном компьютере, подключенном к локальной сети.

5.2.3. Дискový сервер, бездискóвые и другие клиенты локальной сети.

В составе локальной сети ВЦ РАН имеются десять бездискóвых рабочих станций (SPARCstation SLC). Эти рабочие станции включались в сеть как `"diskless-клиенты"`. Для этого при установке одной из рабочих станций (SPARCstation 10/30), вернее при установке на ней операционной системы, эта рабочая станция была выбрана в качестве сервера для бездискóвых рабочих станций. Информация о некоторых бездискóвых станциях задавалась непосредственно при установке операционной системы на дискóвом сервере.

Инсталляция остальных `"diskless-клиентов"` производилась на работающем сервере специальными средствами, в частности, с использованием командного файла `"/usr/etc/install/add_client"`. Для этих целей на основном (sd0) диске сервера было выделено дискóвое пространство (партиции `"d"` и `"e"`) в каталогах `"/export"` и `"/export/swap"`, которые фактически являются корневыми каталогами (партиция `"a"`) и `"swap-пространством"` (партиция `"b"`) для бездискóвых рабочих станций. Эти станции загружают операционную систему через сеть со своего сервера, и после окончания процедуры загрузки отличить эти станции от станций, у которых есть диски, практически невозможно, так как им доступна вся распределенная файловая сетевая система.

На сети была произведена инсталляция рабочей станции (SPARCstation IPC) как `"dataless-клиента"`. Отличие этого типа инсталляции от `"diskless-типа"` заключается в том, что у данной станции должен быть свой, хотя и небольшой по объему диск. На диске создаются партиции `"a"` и `"b"`, т.е. корневой каталог и `"swap-пространство"`, а другой обязательный каталог - `"/usr"` (партиция `"g"`) `"наполняется"` содержимым в результате монтирования соответствующего каталога (файловой системы) на сервере. Таким образом, `"dataless-клиент"` загружает со своего диска свою собственную операционную систему, но в дальнейшем, также как и для бездискóвых станций, его работа полностью зависит от функционирования соответствующего сервера.

Нужно заметить, что для разных `"diskless-клиентов"` и `"dataless-клиентов"` могут быть выделены разные серверы на сети.

5.2.4. Доступ к файловой системе с персонального компьютера.

В ЛВС ВЦ РАН кроме рабочих станций фирмы SUN имеются персональные компьютеры (PC). На них установлено специальное программное обеспечение (PC NFS), позволяющее обеспечивать доступ к распределенной файловой системе ЛВС ВЦ РАН из среды операционной системы персонального компьютера (MS DOS или WINDOWS).

Подключение к сети персональных компьютеров производилось с помощью специальных адаптеров и специального программного обеспечения PC-NFS, которое устанавливалось на каждом компьютере с учетом его технических и программных характеристик (каждый комплект PC-NFS имеет свой уникальный номер, также как и устанавливаемый в персональный

компьютер адаптер). При установке программного обеспечения определялся дисковый сервер (PC-NFS-сервер), на котором запускалась специальная программа - "pcnfsd", которая обслуживает PC-NFS-клиентов со стороны сервера. Кроме этого учитывалось, что в сети работает Сетевая Информационная Система, а для клиента необходим доступ к сетевой файловой системе.

5.3. Пользователи сети, распределение ресурсов и их учет.

5.3.1. Общие положения.

В операционной системе SUN OS имеются широкие возможности для регистрации новых пользователей, выделение для них дискового пространства, учета использованных ресурсов, подключения к сети новых рабочих мест. На каждом узле сети можно иметь локальных пользователей, а можно иметь единственный список пользователей, действительный для каждой рабочей станции сети. Для этого необходимо, чтобы на локальной сети была запущена Сетевая Информационная Система (NIS - Network Information System), которая обеспечивает доступ к специальным базам данных сети. Базы данных содержат информацию о пользователях сети (passwd), об узлах сети (hosts), и другую необходимую для сети информацию. Все узлы, объединенные такой системой, представляют собой поименованную группу машин (domain), а рабочая станция, где располагаются эти базы данных, называется NIS-сервером (или master-сервером).

Для каждого конкретного пользователя на каждой локальной (даже если она общедоступна) файловой системе можно выделить определенное дисковое пространство (квоту), использование которого будет контролироваться операционной системой, т.е. можно наложить ограничения сверху на дисковые ресурсы пользователя в конкретной файловой системе.

После того как создана распределенная файловая система и запущена Сетевая Информационная Система, каждая машина и каждый пользователь в меру своих прав имеет доступ к общесетевым ресурсам.

5.3.2. Регистрация пользователей и выделение дисковых квот.

Для того чтобы зарегистрировать нового пользователя рабочей станции, необходимо выполнить команду (точнее командный файл) `"/usr/etc/install/add_user"`. Здесь же, в командной строке нужно задать несколько позиционных параметров:

- входное имя пользователя (login-name);
- уникальный идентификатор пользователя;
- идентификатор группы;
- полное имя пользователя;
- "домашний" каталог пользователя;
- имя интерпретатора языка shell (sh, csh и т.д.).

В результате выполнения команды модифицируется файл `"/etc/passwd"`, создается "домашний" каталог пользователя на дисковом устройстве, в этот каталог записываются необходимые системные файлы. После этого на данной машине разрешено работать данному пользователю.

Если используется Сетевая Информационная Система, предыдущая процедура выполняется на NIS-сервере, а затем выполняется команда

`"/var/yp/make passwd"`, которая модифицирует NIS-базу данных для сети (вернее NIS-карту `passwd`), и зарегистрированный пользователь становится пользователем всех машин сети, на которых запущена Сетевая Информационная Система. Имеется возможность регистрации локальных пользователей на отдельных машинах. Но в ВЦ РАН эта возможность практически не используется, и супер-пользователь (`root`) является общесетевым (правда с разными правами на сервере и клиентах).

Далее, используя аппарат квотирования, можно выделить пользователю определенное количество дискового пространства. Это пространство (квота) выделяется в той файловой системе, где находится "домашний" каталог пользователя. Квоту можно выделить и в любой другой файловой системе. Но как первое, так и второе можно сделать, если выполнены следующие условия. Во-первых, при конфигурации ядра операционной системы (`vmunix`) должны быть заданы опции `QUOTAS` и `UFS`. Во-вторых, в файле `"/etc/fstab"` (см.5.2) для файловых систем, на которых будет осуществляться квотирование, в соответствующих строках добавить опцию - `"quota"`. И, наконец, в корневых каталогах этих файловых систем необходимо создать пустой файл `"quotas"` (например, командой `"touch quotas"`). После этого можно устанавливать и редактировать размеры квот пользователей с использованием команды `"edquotas"` (как и все привилегированные действия, эту команду может выполнять только супер-пользователь).

При вызове пользовательских квот на редактирование, выдается размер этих квот по всем файловым системам, подвергшимся процедурам, описанным выше. Редактирование производится в терминах текстового редактора `"vi"`, но с дополнительным контролем. Пользователь файловой системы никоим образом не может занять дисковое пространство, большее, чем задано квотированием, хотя здесь предусмотрены как "мягкий" (`soft`), так и "жесткий" (`hard`) лимит.

5.3.3. Сетевая Информационная Система (NIS).

Немного подробнее о Сетевой Информационной Системе (NIS). Вначале выделяется группа машин в сети (`domain`), которая будет иметь свое имя (`domainname`). В этой группе (`NIS-domain`) выделяются один (и только один) `"master"`-сервер, `"slave"`-сервер, дублирующий предыдущий, остальные NIS-клиенты. Затем на каждой машине этой группы создается каталог `"/var/yp"`, а в файл `"/etc/domainname"` записывается имя этой группы. После этого нужно создать на `"master-сервере"` NIS-базу данных, осуществив перепись:

```
"/usr/lib/NIS.Makefile" -> "/var/yp/Makefile"
```

и выполнив команды:

```
"cd /usr/etc/yp; ypinit -m"
```

Последняя команда создает базу данных, в том числе `"ypservers-карты"` (`passwd`-карту, `hosts`-карту и т.д.). И, наконец, производится запуск программ - "демонов", обслуживающих работу NIS:

```
/usr/etc/ypserv
/usr/etc/ypbind
/usr/etc/rpc.yppasswd /etc/passwd -m passwd
/usr/etc/ypxfrd
```

Для того, чтобы после перезагрузки операционной системы все программы - "демоны" перезапускались, в системном файле `"/etc/rc.local"` необходимо сделать соответствующие изменения.

На `"slave-сервере"` последовательность запуска программ следующая:

```

/usr/etc/ypbind
cd /usr/etc/yp; yppinit -s "имя master-сервера"
/usr/etc/ypserv

```

На клиентах (рабочих станциях) запускается только одна программа-"демон":

```

/usr/etc/ypbind

```

Для PC-NFS-клиентов (PC) нужно переконфигурировать PC-NFS-систему с учетом работы в Сетевой Информационной Системе. При конфигурации указать имя "master"-сервера и имя группы машин (domainname).

5.3.4. Учет использования ресурсов на рабочей станции сети.

Для того, чтобы вести учет работы пользователей, использования ресурсов на рабочей станции, SunOS располагает удобными средствами для сбора данных по использованию системы пользователем, процессом с данным номером, данных о сеансе связи и использованию дисков. Эти программные средства носят название ACCOUNTING, и чтобы эти программы исполнялись, необходимы следующие условия.

Во-первых, ядро операционной системы должно быть сконфигурировано с опцией SYSACCT. Во-вторых, должна быть запущена программа "/usr/lib/acct/startup" (в файле "/etc/rc" необходимо иметь соответствующую строку). В этом случае SunOS будет протоколировать в файл "/var/adm/wtmp" информацию о перезагрузках системы, системных командах "shutdown", о начале сеанса работы пользователя, изменении даты и др. В файл "/var/adm/pacct" записывается информация об окончании работы процесса. Чтобы контролировать и обрабатывать эту и другую накопленную информацию необходимо запускать еще несколько программ периодически или в определенное время. Для этого в SunOS имеется специальное средство - CRON (запускается программа-демон "/var/etc/cron"), которое в зависимости от содержимого файла (с правами доступа для суперпользователя) "/var/spool/cron/crontabs/root" запускает те или иные системные программы в заданное время.

Пример файла /var/spool/cron/crontabs/root:

```

0 * * * * /usr/lib/acct/ckpackt
0 1 * * * /usr/lib/acct/dodisk
0 2 * * * /usr/lib/acct/runacct 2 > /usr/lib/acct/nite/fd2log
15 5 1 * * /usr/lib/acct/monacct

```

Используя эти параметры "cron"-демон запускает:

- каждый час (в ноль минут) программу скрасст, отслеживающую размеры файла "/var/adm/pacct",
- каждый день в 1 ночи программу "dodisk", собирающую статистику по использованию дисковой памяти,
- каждый день в 2 часа ночи программу "runacct", суммирующую всю статистику за день, и, наконец,
- каждое первое число месяца в 5 часов 15 минут программу "monacct" - суммирование статистики за месяц.

В результате работы всех этих программ готовится обширная информация, которую можно сразу выдавать на печатающее устройство. Кроме этого можно самостоятельно формировать различные таблицы по тем или иным данным.

5.4. Последовательность работ системного администратора при развертывании сети и ее поддержке.

Здесь мы перечислим основные (далеко не все) функции системного администратора по развертыванию и обеспечению функционирования локальной сети:

- исходя из наличия конкретных рабочих станций, продумывается общая схема сети, определяются функции отдельных узлов сети (серверы и клиенты);

- если в сети предполагается использовать бездисковые рабочие станции, требуется установить одну (дисковую) рабочую станцию как сервер, с учетом закрепления за ним "diskless-клиентов";

- выделяются на сети один или несколько файловых серверов при наличии у них дисковой памяти большого объема. Остальные узлы будут файловыми клиентами;

- если запускается Сетевая Информационная Система (NIS), то определяется имя группы машин (domainname), "master-сервер", "slave-сервер", остальные клиенты системы NIS;

- на каждой рабочей станции форматируются и затем разбиваются на логические все ее дисковые устройства (если это необходимо) и создаются на них локальные файловые системы;

- устанавливается операционная система на каждой рабочей станции ;

- каждому узлу сети присваивается имя, адрес; за каждым пользователем сети закрепляется имя, идентификатор и другие необходимые атрибуты; создается NIS-база данных, периодически дополняемая и модифицируемая;

- на файловых серверах осуществляется экспортирование, а на клиентах - монтирование необходимых файловых систем, что приведет к созданию распределенной сетевой файловой системы. С учетом требований пользователей и системных нужд эти возможности регулируются и модифицируются;

- каждому пользователю сети выделяется дисковая квота на одной или нескольких файловых системах, производится изменение этих квот и контроль за их использованием;

- на рабочих станциях запускается система учета прохождения задач и работы пользователей в сети - ACCOUNTING;

- на файловом сервере (или на каком-либо другом) устанавливаются общедоступные системы программирования (трансляторы, графические пакеты, информационные системы и др.);

- для обеспечения выхода с узлов локальной сети в глобальную сеть (Internet) выполняются следующие организационные работы: запрашиваются Internet-адреса, имя DOMAIN'a, согласовываются возможности выхода на "DOMAIN'ы" более высокого уровня; кроме того, после установления действительных физических связей нижнего уровня запускаются соответствующие компоненты сетевого программного обеспечения (реконфигурируется ядро операционной системы, устанавливаются соответствующие драйверы, запускается служба DNS - Domain Name Service).

После того, как сеть создана и эксплуатируется, она требует постоянного пристального внимания со стороны службы администрирования.

Текущая работа административной службы состоит из:

- регистрация новых пользователей, выделение для них необходимых ресурсов;

- подключение новых узлов к сети, реконфигурация уже подключенных узлов, подключение другого оборудования, установка новых компонент программного обеспечения;

- организация надежной работы файловой системы (служба "backup", "ремонт" файловых систем);

- сбор информации о работе системы через визуальный контроль, мнение пользователей и автоматические средства;
- проведение на базе собранной информации мероприятий по оптимизации работы системы (реконфигурация сети, передислокация системных программных компонент);
- проведение консультаций и обучение пользователей.

Заключение.

В настоящее время завершен первый этап развертывания ИВС ВЦ РАН). Полностью функционируют базовые средства ЛВС ВЦ РАН. Это значит, что в рамках локальной сети, построенной на протоколах ETHERNET и TCP/IP, функционируют:

- физическая среда, развернутая на структурированной кабельной сети с витыми парами (UTP) по звездной топологии;
- центр коммутации на базе интеллектуального коммутирующего устройства 3Com Linkbuilder 3GN;
- корпоративная файловая система, реализованная на основе SPARCstation 10, дисковых массивов типа RAID класса 5 фирм VoxHill и SSI с суммарной емкостью 23 Гбайта и других дисковых накопителей;
- серверная рабочая станция SPARCstation 10/30;
- 64-процессорная транспьютерная система параллельной обработки данных GCel 1/64 с собственным Frontend компьютером SPARCstation 2;
- 10 рабочих станций типа SLC;
- 30 лицензированных точек доступа к сети от PC с использованием адаптеров Etherlink III.

Кроме того в настоящее время завершаются работы по установке терминального сервера фирмы SUN Microsystems с 16-канальным модемным узлом RS-1600 фирмы ZyXEL для обеспечения удаленного доступа к системе через коммутируемые и выделенные линии. Также в ближайшее время в качестве мощного вычислительного и графического сервера в систему будет включена рабочая станция DEC 3000/300X фирмы Digital Equipment Corporation.

Теледоступ к ресурсам и программным системам обеспечивается как непосредственно с рабочих станций сети, так и с PC.

Базовое программное обеспечение предоставляет возможность традиционной работы пользователей в канонических системах программирования (C, FORTRAN-77, PASCAL), используя как последовательные (SPARCstation), так и параллельные архитектуры (GCel 1/64).

На очереди широкое внедрение новых информационных технологий в концепции, изложенной в разделе 4. Началом этой большой работы являются уже освоенные пользователями графические системы (пока в рамках традиционных технологий) и внедрение разнообразных freeware-продуктов. Авторский коллектив предполагает в будущем посвятить выпуск наших трудов именно развитию информационных технологий.

Базовые средства ЛВС ВЦ РАН также будут модифицироваться, и этот процесс практически не прекращается. Как уже было изложено в гл.1, следующий этап развития включает в себя выход к магистральным линиям связи (backbone) и выход на космические (спутниковые) линии связи. Безусловно, реализация этих двух важнейших направлений уже выходит за пределы финансовых возможностей ВЦ РАН и может быть осуществлена только в рамках Федеральных программ развития фундаментальных и прикладных исследований в Академии наук РФ. Тем не менее, эти возможности заложены в базисные основы проекта и согласованы с общей концепцией построения ИВС.