Глава 3.
Проектирование архитектуры кластерной системы

Если говорить совсем кратко, то вычислительный кластер – это совокупность компьютеров, объединенных в рамках некоторой сети для решения одной задачи. В качестве вычислительных узлов обычно используются доступные на рынке однопроцессорные компьютеры, двух или четырехпроцессорные SMP-серверы. Каждый узел работает под управлением своей копии операционной системы, в качестве которой чаще всего используются варианты стандартных ОС: Linux, Windows, Solaris и другие. Состав и мощность узлов могут меняться даже в рамках одного кластера, что дает возможность создавать неоднородные системы. Выбор конкретной коммуникационной среды определяется многими факторами: особенностями класса решаемых задач, доступным финансированием, необходимостью последующего расширения кластера и т. п. Часто включают в конфигурацию кластера специализированные компьютеры, например, файл-сервер. Как правило, предоставляется возможность удаленного доступа на кластер через Интернет.

Ясно, что простор для творчества при проектировании кластеров огромен. Узлы могут не содержать локальных дисков, коммуникационная среда может одновременно использовать различные сетевые технологии, узлы не обязаны быть одинаковыми и масса прочих нюансов. Рассматривая крайние точки, кластером можно считать как пару ПК, связанных локальной сетью Fast Ethernet, так и рекордные вычислительные системы из первых строк списка 500 самых мощных систем мира, объединяющих тысячи процессоров.

Рассмотрим аппаратную сторону будущей кластерной системы. Возможны различные варианты критериев, с позиций которых будут оцениваться принимаемые в дальнейшем решения. Однозначно

посоветовать что-либо здесь просто невозможно, все определяется задачами каждого конкретного проекта. Часто при заданном бюджете требуется получить максимальную производительность системы на решении некоторого класса задач. Другой вариант – минимизировать стоимость проекта при заданной производительности. Для кого-то важна компактность системы, для других ее отказоустойчивость и высокая степень доступности. В последнее время очень важной характеристикой становится энергопотребление, которую рассматривают либо в абсолютном исчислении, либо принимают в расчет отношение производительности системы к ее энергопотреблению. Вариантов оценки и выбора существует много, поэтому, начиная проектирование кластера, определитесь, что первично, а чем в какой-то степени можно и пожертвовать.

Рассмотрим базовые компоненты и их характеристики, которые должны учитываться при построении кластерных систем:

- размещение и компоновка кластера,
- вычислительные узлы,
- управляющий узел,
- файл-сервер и хранилище данных,
- сетевая инфраструктура,
- источники бесперебойного питания.

В процессе проектирования важно осознать, что сейчас для конструирования кластерной системы в руки дается мощный конструктор, с помощью которого можно собрать именно ту систему, которая в наибольшей степени будет соответствовать решаемым задачам и бюджету проекта. Нужно правильно сформулировать свои пожелания и вовремя высказать их фирме-поставщику.

Начнем с того, что для эффективного обслуживания кластера не последнюю роль будет играть его компоновка. Лучшее решение – это **расположение кластера в стойке**. Даже для небольшого кластера из 4-6 узлов стойка уже уместна (рис. 3.1), при этом увеличение стоимости решения будет не столь существенным.

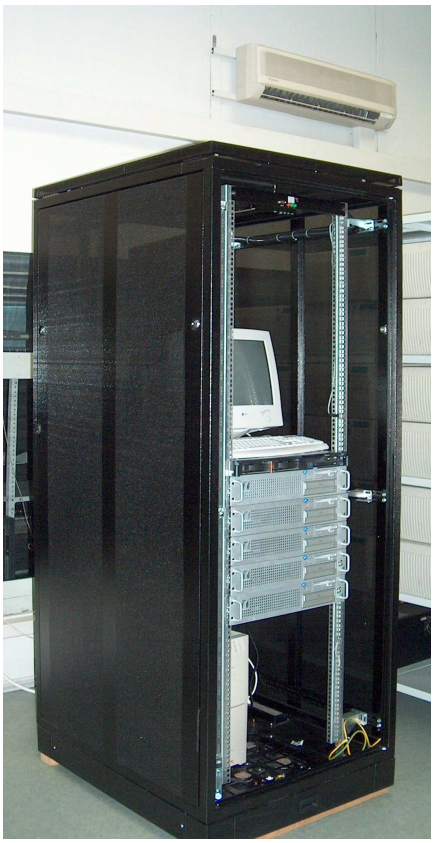

Рис. 3.1. Пример стойки с пятью узлами и монитором

Не стоит сильно экономить на собственно стойке, так как недоработки ее конструкции могут в последствии превратиться в неожиданные проблемы. При выборе стойки следует обратить внимание на следующие нюансы.

- Соответствие формата стойки формату узлов кластера (обычно это стандарт 19-дюймовой стойки).
- Наличие в комплектах узлов, головного узла, файл-сервера и сетевых коммутаторов рельсов (rail kit) для крепления в стойку.
- Соответствие крепежа рельсов и стойки. На практике используются два основных стандарта, которые называют Compaq и HP. В первом случае отверстия для крепления в профилях стойки будут круглыми, во втором – квадратными. Переходников с одного стандарта на другой не существует.
- Наличие кабельных органайзеров. К каждому вычислительному узлу кластера будут приходить как минимум питание и сеть, поэтому представьте, что будет твориться в местах скопления этих проводов.
- Перфорированная лицевая дверь стойки. Если поставить сплошную дверь, например, стеклянную, то обеспечить полноценный доступ

Рис. 3.2. Пример стойки с 30 узлами формата 1 U

холодного воздуха к узлам будет невозможно, однако почти все современные стоечные серверы захватывают воздух для охлаждения только спереди.

Расположить стойку в помещении нужно так, чтобы был удобный доступ к узлам и спереди, и сзади. Спереди должно быть достаточно места для того, чтобы поместился узел из стойки и человек. Сзади будет выходить тепло, поэтому расстояние до стены должно быть достаточным для его рассеивания.

Обратите внимание, что стоечный крепеж обычно поставляется отдельно, поэтому, приобретая стойку, купите там же соответствующие винты и гайки. Стоит закупить их с запасом, чтобы сберечь свое время в будущем.

Не менее важным, чем выбор стойки, является **выбор форм-фактора вычислительных узлов**. Устоявшихся решений в настоящее время существует несколько.

Стоечное решение с серверами 3-4 U. Обслуживание такого решения, наверное, самое легкое. Любое дополнительное периферийное оборудование устанавливается в такие серверы без особых проблем, охлаждение в них производится легко. Основным недостатком этого решения является то, что много таких узлов в стойку не войдет.

Стоечное решение с серверами 1-2 U (рис. 3.2)**.** Вариант компактен, таких узлов в стойку войдет уже больше, чем в предыдущем

случае, однако установить в них какую-то дополнительную плату иногда бывает весьма сложно. Тщательно продумайте конфигурацию узлов. Охлаждение в серверах небольшого размера, как правило, организуется сложнее, поэтому еще раз проверьте параметры и общий план разворачивания системы кондиционирования. Холодный воздух должен подаваться к передней части стойки, и кондиционер не должен располагаться слишком близко.

Стоечное решение на блейд-серверах (лезвиях). Самое компактное решение, поэтому и стоимость его при прочих равных условиях немного выше. Управление блейд-серверами обычно организуется с помощью специальных средств, которые нужно будет дополнительно изучить. Такие серверы, как правило, ограничены в плане расширения, и далеко не всегда в них можно установить необходимую новую плату. Например, если штатная коммуникационная сеть на базе Gigabit Ethernet не устраивает, то стоит задуматься о целесообразности такого решения. Некоторые компании предлагают на выбор несколько вариантов сетевого комплектования.

В последнее время появились компактные и исключительно мощные стоечные кластерные решения. Например, компания Rackable Systems (`http://www.rackable.com`) в конце 2006 года анонсировала выпуск кластерных систем, построенных на основе стоечных серверов размера 1 U и четырехядерных процессоров Intel Xeon серии 5300. В одной стандартной стойке на 42 U может быть

Рис. 3.3. Пример расположения кластера из 16 узлов на стеллажах

собран кластер с пиковой производительностью 6,5 Тфлопс.

Немного особняком стоит вариант, в котором *обычные корпуса серверов располагаются на стеллажах* (рис. 3.3)**.** В этом случае кластер займет значительно больше площади, чем стоечное решение. Оборудование обойдется дешевле, однако обслуживание кластера в будущем станет организационно сложнее, а его развитие будет сильно ограничено.

Если число вычислительных узлов невелико, то может быть полезен KVM – переключатель, позволяющий использовать одну клавиатуру и один монитор для всех узлов кластера. Более перспективным и удобным решением является использование сервисной сети (о ней будет рассказано ниже), в этом случае переключатель, скорее всего, не потребуется.

В некоторых случаях идут на совмещение функций вычислительного кластера и отдельных рабочих мест (учебного класса). На каждый узел ставится полный комплект: монитор, клавиатура, мышь, а разграничение функций производится административно, чаще всего, по времени: днем оборудование используется в режиме отдельных рабочих мест, а ночью в режиме кластерной вычислительной системы.

Выбор архитектуры и параметров вычислительных узлов, во многом, определит характеристики будущего кластера. Тип и частота процессоров, свойства чипсета, частота системной шины, объем и частота оперативной памяти, ее конструктив, параметры кэш-памяти, состав портов и поддержка периферии, возможности для будущей модернизации узлов – со всем этим нужно определиться сейчас. Желательно хотя бы приблизительно представить круг задач, которые будут решаться на кластере, чтобы подбирать состав вычислительных узлов, уже исходя из требований, предъявляемых этими задачами.

Основа – это **выбор процессора**, который будет диктовать многие дальнейшие шаги. Одни вычислительные задачи оптимизированы под процессоры Intel Xeon EM64T, какие-то приложения показывают хорошие

характеристики на AMD Opteron, для кого-то очевидные преимущества окажутся у Intel Itanium-2, а в каких-то случаях программа оптимально использует технологию Hyper-Threading. Самым лучшим и правильным вариантом является предварительное тестирование узлов на типичных приложениях, которое покажет реально достижимые в каждом случае параметры. В некоторых случаях узлы на тестирование может предоставить потенциальный поставщик кластерной системы либо сама компания-производитель процессоров. Если будет использоваться новое оборудование или же компоненты в нестандартной конфигурации, то предварительное тестирование нужно выполнить в обязательном порядке либо заручиться гарантийными обязательствами со стороны поставщика. Если есть хотя бы малейшие сомнения в достижении заявленных параметров, то обязательно проконсультируйтесь с профессионалами. А если совсем честно, то проконсультироваться лучше в любом случае.

В последнее время все производители процессоров переходят на многоядерные технологии, что нужно обязательно учитывать. Выбирая одноядерные модели процессоров, можно значительно сэкономить в цене на процессоры, однако больших перспектив это решение не имеет. Вычислительный мир стал параллельным, число ядер на кристалле будет только увеличиваться. Мы уже говорили о том, что первый основной толчок к массовому использованию параллельных вычислительных технологий дали кластерные системы. Окончательный переход к параллелизму определила многоядерность современных процессоров. Подумайте еще раз, настолько ли важна сиюминутная экономия, чтобы отказываться от будущих технологий.

Останавливаясь на многоядерном варианте, проверьте, что эта пока еще новая особенность поддерживается на всех уровнях ПО: в компиляторах, библиотеках, в необходимых прикладных пакетах и системах. Есть ли смысл вкладываться в дорогую аппаратуру, если потом ее нельзя будет эффективно использовать? Однако при всех “за” и “против” различных вариантов, хотим подчеркнуть еще раз: **в конечном итоге все**

определяется способностью кластера решить задачи проекта. Если с помощью такой-то конфигурации кластера задачи будут решены оптимально, то этот вариант и будет хорошим.

При проектировании архитектуры узлов обратите внимание на то, что иногда с увеличением тактовой частоты процессоров растет не только их производительность и стоимость. Для некоторых моделей это ведет к росту энергопотребления, следовательно, выбор более мощных узлов может потребовать перерасчета необходимых характеристик по электричеству и охлаждению для кластерного проекта в целом. Например, можно остановиться на выборе двуядерных процессоров Intel Xeon серии 5100 (Woodcrest) с тактовыми частотами 1,6 ГГц, 1,86 ГГц, 2 ГГц, 2,33 ГГц, 2,66 ГГц и 3 ГГц. При этом энергопотребление всех моделей составляет 65 Вт, а у старшей модели – 80 Вт. Аналогично у четырехядерного процессора Intel Xeon серии 5300 (Clovertown): энергопотребление моделей на 1,6 ГГц, 1,86 ГГц, 2,33 ГГц равно 80 Вт, а у старшей модели на 2,66 ГГц – 120 Вт. Четырехядерный процессор AMD Barcelona будет выпускаться в нескольких версиях, различающихся по тепловым характеристикам: основная модель будет выделять 95 Вт, низковольтовая – 68 Вт, а высокопроизводительная – 120 Вт.

Для многих задач критически важным является **объем и скорость работы оперативной памяти**. В этом случае лучше пожертвовать одним-двумя вычислительными узлами в пользу покупки памяти с требуемыми характеристиками. Если рассматривается вариант возможного увеличения объема оперативной памяти в будущем, то обратите внимание на конструктив материнской платы и число свободных слотов для дополнительных модулей.

Схожий и весьма важный вопрос – **структура и объем кэш-памяти**. В зависимости от типа процессора эти характеристики могут сильно меняться, влияя на эффективность работы конечных приложений. При этом важны не только количественные показатели, но и такие свойства, как разделение кэш-памяти последнего уровня и доступа к

системной шине между отдельными ядрами процессора. Например, двуядерный процессор Intel Itanium-2 Montecito/9050 имеет по 12 Мбайт кэш-памяти третьего уровня на каждое ядро (24 Мбайт на кристалл), однако оба ядра разделяют один канал доступа к системной шине процессора. На практике, соотношение между локальностью использования данных и локальностью вычислений в программе, степенью пересечения между ядрами по используемым данным для каждого конкретного приложения определят реальный выигрыш как от структуры и иерархии памяти, так и от метода доступа к общим ресурсам каждого конкретного процессора.

Наличие **локальных дисков** на узлах кластера, на первый взгляд, кажется излишним. В самом деле, зачем тратить лишние деньги? Эти диски работать почти не будут, загрузку ОС можно сделать по сети, да и ее обновления в будущем делать будет проще. Примерно так нередко и рассуждают при построении кластера.

Верно, однако отсутствие локальных дисков влечет за собой и потенциальные проблемы, о которых сначала не всегда задумываются. Например, если возникнут сетевые проблемы или же проблемы с сервером DHCP/NFS/TFTP, то их решение и даже определение причин будет изрядно затруднено. Опять-таки, как только задача исчерпает всю оперативную память и затребует еще, ее просто снимут, поскольку области свопинга нет. Есть возможность использовать swap-over-nfs, но в этом случае скорость работы задачи упадет в десятки, а то и сотни раз. Более того, может существенно снизиться скорость работы других приложений, так как нагрузка на общий NFS-сервер, естественно, отразится на всех узлах. При этом цена обсуждаемого вопроса по сравнению со стоимостью самого узла крайне невысока. Достаточно поставить на узел один IDE- или SATA-диск минимального объема для того, чтобы в будущем избежать массы проблем.

Еще одним весомым аргументом за включение дисков в состав узлов является возможность локализации ввода/вывода для определенного класса приложений. Если этого не предусмотреть, то все файловые

операции программ будут идти через файл-сервер и тот или иной вариант сетевой файловой системы, что медленнее, а иногда и намного медленнее, чем использование жестких дисков на самих узлах. Это особенно касается приложений класса Out-of-Core, работающих с данными, объем которых намного превосходит объем суммарной оперативной памяти компьютера в целом. Единственным вариантом эффективной работы таких приложений является аккуратная организация обменов с дисками для подкачки новых данных в память, что должно проходить на фоне и за время обработки данных, уже лежащих в памяти.

Floppy- и CD/DVD-приводы на узлах используются редко, в основном для начальной установки операционной системы. Некоторые модели серверов позволяют использовать виртуальные приводы, подключаемые по сети с головного узла. Если такой возможности не предусмотрено, можно воспользоваться приводом, подключаемым через USB. В этом случае узлы должны поддерживать USB и загрузку с подключаемых приводов, а использованный в них USB-контроллер должен поддерживаться выбранной операционной системой.

Если планируется использовать платы ServNet или аналогичные для организации сервисной сети, то может потребоваться наличие в узлах COM-порта.

Говоря об общей архитектуре, отметим, что кроме вычислительных узлов, необходимо предусмотреть **головной узел кластерной системы**. На головном узле пользователи компилируют свои программы, готовят данные для счета, проводят предварительную обработку данных. С головного узла производится запуск задач.

Совмещать головной узел с одним из вычислительных узлом не рекомендуется, так как и работа пользователей будет затруднена, и эффективность параллельных программ, распределенных на такой узел, будет ниже. Вместе с тем, в некоторых случаях подобное совмещение вполне оправдано и само по себе не может рассматриваться дефектом

проекта: все определяется исключительно режимом использования будущей кластерной системы и стоящими задачами.

Так как на головном узле, скорее всего, будут работать несколько пользователей одновременно, то дополнительная оперативная память на нем лишней не будет. Для ускорения работы локальных приложений стоит установить в него быстрый жесткий диск или рейд-контроллер. К процессору головного узла больших требований, как правило, не предъявляется, вычислительные задачи на нем работать не будут, а для компиляции большой мощности не нужно. Если заранее известно, что пользователей будет много, то имеет смысл использовать компьютер с несколькими процессорами, в противном случае и это не является обязательным требованием.

Весьма тонкий момент, который следует продумать заранее, это использование на головной машине процессора, отличного от процессоров вычислительных узлов. Часто компиляторы устроены так, что выполняют различного рода оптимизацию кода именно под тот процессор, на котором идет собственно процесс компиляции. Эта проблема, как правило, легко решается, поскольку многие компиляторы поддерживают режим кросс-компиляции, генерируя код целевого процессора, однако убедиться в эффективности и целесообразности такого режима работы нужно до заказа оборудования.

Кроме головного узла, в составе кластера **необходим файл-сервер**. Его функции могут быть переложены на головной узел, если предполагаемые нагрузки на сетевой диск будут не очень большими.

На файл-сервере стоит предусмотреть аппаратный рейд-контроллер. Оптимальным является использование RAID-5 или RAID-6, так как в этом случае надежность, а часто и скорость работы повышаются по сравнению с одиночным диском. Выбирая рейд-контроллер, обратите внимание на следующие параметры:

- максимальное количество подключаемых дисков,
- число дисков, использующихся для контроля четности в RAID-5,

- возможность “горячей” замены дисков,
- поддержка контроллера операционной системой.

В настоящее время большинство серверных материнских плат имеют интегрированный рейд-контроллер, но к нему не всегда можно подключить более 2-4 дисков. На рынке подобные решения представлены исключительно широко: от внутренних плат рейд-контроллеров, до аппаратных файл-серверов.

Если предполагается использовать RAID-1, -5 или -6, то полезно подумать о запасных (spare) дисках. Они не используются до тех пор, пока основные диски работают нормально. Как только один из жёстких дисков выходит из строя, вместо него подключается запасной. На него записывается вся необходимая информация, которая восстанавливается с работающих носителей за счёт изначальной избыточности хранения (поэтому такие диски не столь полезны в RAID-0), и работа RAID-массива продолжается в нормальном режиме. На один RAID-массив стоит выделить хотя бы один такой диск. Убедитесь, что RAID-контроллер поддерживает работу с запасными дисками и умеет включать их в работу автоматически, без остановки работы массива.

Сетевая инфраструктура в современных кластерных системах представлена тремя вариантами сетей: коммуникационная, транспортная и сервисная. Во многих проектах все три сети присутствуют одновременно.

Коммуникационная сеть – это тот компонент, который во многом определит эффективность работы программ на будущем кластере, поскольку именно с помощью этой сети процессы параллельных программ обмениваются данными между собой. Какими характеристиками выражается производительность коммуникационных сетей в кластерных системах? Для работы параллельных приложений в наибольшей степени важны две характеристики: латентность и пропускная способность сети. Латентность – это время, необходимое для передачи сообщения нулевой длины от одного процесса параллельной программы другому. Пропускная способность сети определяет собственно скорость передачи информации

по каналам связи. Если в программе много маленьких сообщений, то на эффективность ее выполнения сильно скажется латентность. Если сообщения передаются большими порциями, то важна высокая пропускная способность каналов связи. Заметим, что из-за латентности максимальная скорость передачи по сети не может быть достигнута на сообщениях с небольшой длиной, а насколько быстро реальные показатели приближаются к заявленным характеристикам – это будет определяться и типом сети, и набором оборудования от конкретного производителя. На практике не столько важны указанные производителем пиковые характеристики коммуникационной аппаратуры, сколько реальные показатели, достигаемые на уровне приложений пользователей, в частности, на уровне MPI-программ или прикладных пакетов.

Выбор коммуникационной сети полностью определяется свойствами решаемых задач (выполняемых программ) и целями создания кластерной системы. Если обменов между параллельными процессами приложения мало, то достаточно использовать какой-либо вариант Ethernet. В зависимости от объемов пересылок это 100 Мбит/с или 1 Гбит/с. Серьезным недостатком сети Ethernet является высокая латентность, составляющая около 130-150 мкс на пакет. Некоторые современные коммутаторы позволяют снизить этот показатель до 50-60 мкс, однако для многих вычислительных задач это все равно не решает проблемы.

В данный момент на рынке доступны несколько стандартов сетевого оборудования, дающего скорости более 1 Гбит/с, и со значительно более низкой латентностью. Рассмотрим наиболее распространенные и доступные варианты.

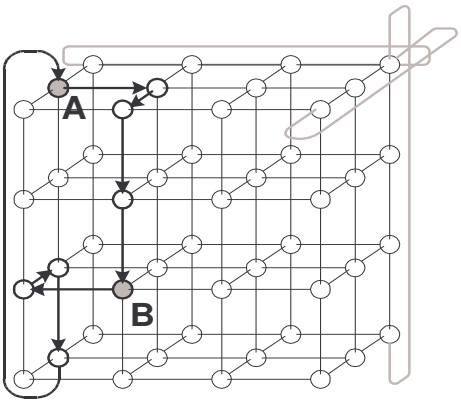

Рис. 3.4. Соединение узлов в трехмерный тор в сети SCI

Сетевая технология SCI. Скорость передачи данных около 700 Мбайт/с (5,6 Гбит/с), аппаратная латентность на уровне 0,2 мкс. На MPI-приложениях скорость передачи данных достигает значений около 350 Мбайт/с, а латентность – 1,4 мкс. Особенностью данной сетевой технологии является то, что она не требует коммутатора. Все сетевые адаптеры соединяются в кольцо либо тор (поддерживаются двумерные и трехмерные топологии). На рис. 3.4 показан пример объединения машин кластера с топологией трехмерного тора. Два кластера суперкомпьютерного комплекса НИВЦ МГУ использовали технологию SCI, объединяя узлы в соответствии со структурой двумерного тора. Кластер СКИФ-К500, установленный в ОИПИ НАНБ, использует технологию SCI с объединением 64 узлов в трехмерный тор.

С одной стороны такая архитектура кажется выгодной, так как не нужно приобретать дополнительного оборудования, только сетевые адаптеры. При добавлении новых узлов также не надо заботиться о покупке нового коммутатора. С другой стороны, опыт показывает, что эта технология весьма трудоемка при первоначальной настройке. Более того, находить ошибки или сбои в коммутации такой сети крайне сложно. Средств для ее диагностики существует очень мало. Но самым неудобным является то, что выход их строя одного вычислительного узла означает отключение двух (для двумерного тора) или трех (для трехмерного) колец связи. Сбой двух узлов в двумерном торе приведет к отключению еще двух

других узлов, лежащих на пересечении отключенных колец. Для трехмерного тора аналогичное отключение произойдет при сбое трех узлов.

Сетевая технология Myrinet-2000/Myri-10G. Скорость передачи данных – 2 Гбит/с и 10 Гбит/с соответственно. Для MPI-приложений достигается скорость 247 Мбайт/с и 1,2 Гбайт/с, латентность – 2,6 мкс и 2 мкс. Данная технология использует оптоволоконные соединения. При построении сети используются коммутаторы, которые можно соединять между собой. Драйверы и программное обеспечение распространяются бесплатно. Стандарт Myrinet существует довольно давно, поэтому широко поддерживается. Технология Myrinet в течение уже многих лет активно используется в Межведомственном суперкомпьютерном центре РАН для построения нескольких поколений суперкомпьютерных кластерных систем.

Сетевая технология InfiniBand. Скорость передачи данных до 10 Гбит/с, латентность – менее 1 мкс. На MPI-приложениях скорость передачи данных в дуплексном режиме достигает 2,5 Гбайт/с, латентность – 1,3-1,7 мкс. Реальная скорость передачи в настоящее время ограничивается скоростью шин PCI-X и PCI-Express. В случае PCI-X адаптеров скорость составит не более 1 Гбайт/с. Не так давно анонсировано сетевое оборудование, построенное по этой же технологии, со скоростью передачи данных до 30 Гбит/с.

Технология InfiniBand разработана альянсом ведущих мировых разработчиков: Intel, IBM, Cisco, Sun и рядом других. В настоящее время на компьютерном рынке представлено множество брендов, поставляющих это оборудование, например, только что упоминавшаяся компания Cisco, Qlogic или Mellanox.

Данная технология одна из самых молодых среди аналогов, но и одна из самых быстро развивающихся. При построении сети используются коммутаторы, которые можно объединять между собой. Аппаратно поддерживается маршрутизация пакетов, при этом дублирующиеся

маршруты используются для балансировки нагрузки.

При соединении нескольких коммутаторов часто используется топология “Fat-Tree”. В этом случае коммутаторы соединяются несколькими каналами параллельно, что увеличивает скорость канала между ними. Базовая поддержка InfiniBand уже есть в последних ядрах Linux. Существуют библиотеки MPI и программное обеспечение для работы с InfiniBand, распространяемые бесплатно.

В настоящее время коммуникационные технологии развиваются очень быстро. Недавно появились и уже сегодня доступны на рынке технологии Quadrics и 10 Gbit Ethernet, представлены опытные образцы технологии 100 Gbit Ethernet.

Кроме коммуникационной сети, по которой будут обмениваться информацией параллельные приложения, рекомендуем отдельно поставить **транспортную сеть** (иногда ее называют управляющей сетью). По ней происходит передача данных сетевой файловой системы и может осуществляться управление вычислительными узлами.

Разделение коммуникационной и транспортной сетей необходимо для того, чтобы вспомогательный трафик не мешал работе параллельных приложений. Даже если в качестве коммуникационной сети используется Ethernet, поставьте второй коммутатор и настройте на нем независимую транспортную сеть. Затраты небольшие, а в трудных ситуациях это поможет не только спасти данные, но и значительно облегчить работу администратору и избежать долгих часов простоя системы.

Еще одна сеть, о которой нужно обязательно упомянуть, это **сервисная сеть**. Эта сеть используется исключительно для обслуживания вычислительных узлов. Оконечным оборудованием для этой сети должен быть, строго говоря, даже не вычислительный узел, а устройство, способное его контролировать. На многих современных серверных платформах такие устройства устанавливаются изначально. Эти устройства доступны и отдельно, например, платы Hewlett-Packard Lights-Out или платы ServNet, разработанные в Институте программных систем РАН. Все

они предоставляют различный уровень сервиса от минимальной возможности удаленно перезагрузить узел, включить или выключить питание, до получения графической консоли.

Крайне важной может оказаться информация с системной консоли при диагностике сбоев: узел завис, и получить доступ к консоли штатными средствами невозможно. Большинство имеющихся на рынке устройств доступ к системной консоли дают. Например, плата ServNet позволяет получить не только доступ к консоли через стандартный COM-порт, но также управлять питанием узла. Такой функциональности вполне достаточно для решения большинства задач по удаленному обслуживанию кластера.

Получить дополнительную информацию об упомянутых технологиях можно на сайте `http://skif.pereslavl.ru/` (поиском слова ServNet) и поиском по ключевым словам “Lights-Out 100 Remote Management Card” на сайте `http://www.hp.com`.

Наличие сервисной сети сильно упрощает обслуживание кластера. Администратору не надо идти в другое помещение для того, чтобы перезагрузить зависший узел, можно быстро выяснить причину зависания, а в некоторых случаях даже переустановить ОС прямо со своего рабочего места.

Очень важный момент, про который забывают или на котором часто возникает желание сэкономить, – это обеспечение качественного электропитания. От этого будет зависеть и долговечность работы кластера, и степень его доступности, и работоспособность критических приложений. Включить в конфигурацию **источник бесперебойного питания** стоит хотя бы для того, чтобы элементарно защитить вычислительные узлы. Корректное выключение кластера в случае аварийного выключения питания поможет спасти данные долгих расчетов, а очень часто и значительное время, необходимое на восстановление работоспособности кластера.

На практике чаще всего возникают кратковременные скачки

напряжения, с чем любой UPS прекрасно справится. Более серьезный вариант предполагает, что UPS сможет обеспечить работу кластера в течение 5-10 минут. Рассчитать мощность UPS можно исходя из мощности подключенного к нему оборудования. Для работоспособности приложений во время непредвиденных скачков по питанию не забудьте подключить к UPS не только вычислительные узлы, но и все сетевые коммутаторы.

Обязательно завершите проектирование **составлением подробной схемы кластерной системы**, показывающей архитектуру комплекса, его составные части, детали компоновки, особенности размещения, структуру коммуникаций. Не забывайте в будущем модифицировать схему сразу после модернизации кластера, поскольку даже очевидные действия со временем забываются, и много времени будет уходить на восстановление и правильное описание текущей конфигурации.

И еще раз повторим вопрос, поставленный в конце предыдущего раздела. Будет ли кластер расширяться или как-то модифицироваться в будущем? Финансирование проекта часто появляется порциями и возникает большое желание сначала купить часть системы, а потом ее нарастить еще каким-то числом узлов. К такому сценарию развития кластерного проекта подталкивает и тот факт, что архитектуры кластерных систем хорошо масштабируются, поэтому принципиальных препятствий для расширения нет.

Если выбирается вариант последовательного развития кластерной системы, то, как правило, выбирают два пути. Первый предполагает добавления какого-то числа новых вычислительных узлов в существующую систему. Основная проблема здесь одна: если возможность для расширения появилась через полгода-год после создания первой версии системы, то возникает вопрос о целесообразности покупки уже устаревшего оборудования. Приобретать такие же узлы уже не хочется, так как на рынке по схожей же цене есть более привлекательные модели. Но если купить современные варианты узлов, то кластер перестанет быть однородным, и возникнут очевидные проблемы с его использованием.

Второй путь развития – это полная замена каких-либо компонентов системы на новые аналоги: более мощные процессоры, больший объем памяти, новые диски, новые материнские платы с увеличенной частотой системной шины и т.п. И по такому пути вполне можно двигаться, но постарайтесь сразу ответить на сакраментальный вопрос: что делать со старыми комплектующими?

Любое оборудование, даже фирменное и самое дорогое, ломается. Не пренебрегайте изучением **гарантийных обязательств** производителя, а также его **технической поддержкой** и **сервисным обслуживанием**. Эти позиции можно и нужно включать в проект. Гарантия производителя позволит заменить сбойную деталь или целый узел в случае брака, но на это может уйти несколько дней или даже недель. Техническая поддержка позволит избавиться от необходимости искать самим сбойный компонент и перепоручить эту обязанность профессионалам. Особенно удобно, если предусмотрена поддержка “on-site”, когда специалист выезжает прямо на место установки кластера. Сервисное обслуживание позволит проводить не только обмен сбойных компонент, но и ремонт узлов.

Наличие гарантии, безусловно, дает уверенность в том, что сбойный компонент будет заменен. Однако оно не дает уверенности в том, что он будет заменен быстро. Если приложению необходимы все процессоры, а один из них вышел из строя, то дело встанет. Придется ждать замены, которая займет от нескольких дней, до нескольких недель в зависимости от гарантийных обязательств поставщика и производителя оборудования.

Чтобы избежать подобной неопределенности хорошим решением станет **резервный узел**. Его можно заранее сконфигурировать, но не включать в общее счетное поле. При выходе одного из узлов из строя его можно заменить резервным на время ремонта. Это особенно важно для тех кластерных систем, в которых узлы расположены по некоторой жестко заданной структуре, а выход из строя одного узла может нарушить структуру целого сегмента (например, SCI-кластеры с топологией

двумерного или трехмерного тора).

Другое решение – это **резервные запчасти**. Вместо целого узла можно приобрести в резерв только наиболее часто ломающиеся детали, в частности, жесткие диски, вентиляторы для охлаждения, модули памяти, блоки питания, а также запасные коммуникационные карты.

По какому пути обеспечения резерва пойти в каждом конкретном проекте: резервные узлы, компоненты или же вообще отказаться от резерва, – это предстоит решить руководству каждого кластерного проекта самостоятельно. Решение определяется и бюджетом, и стоящими задачами, и требуемой степенью доступности кластерной системы. Что-либо сказать априори одинаково хорошо подходящее для любого случая здесь практически невозможно.

Всегда хочется как-то проверить принятые решения и сравнить с чем-то уже работающим, поэтому в **Приложении 1** приведено несколько примеров программно-аппаратных конфигураций реально существующих кластерных систем. Множество других вариантов можно найти на страницах информационно-аналитического центра Parallel.ru, а также на сайте `http://www.supercomputers.ru` списка Top50 самых мощных компьютеров СНГ – большинство из них являются кластерами.