RAID Levels

Совершенствуя системы хранения данных

Перенос центра тяжести с процессоро-ориентированных на дата-ориентированные приложения обуславливает повышение значимости систем хранения данных. Вместе с этим проблема низкой пропускной способности и отказоустойчивости характерная для таких систем всегда была достаточно важной и всегда требовала своего решения.

В современной компьютерной индустрии в качестве вторичной системы хранения данных повсеместно используются магнитные диски, ибо, несмотря на все свои недостатки, они обладают наилучшими характеристиками для соответствующего типа устройств при доступной цене.

Особенности технологии построения магнитных дисков привели к значительному несоответствию между увеличением производительности процессорных модулей и самих магнитных дисков. Если в 1990 г. лучшими среди серийных были 5.25" диски со средним временем доступа 12мс и временем задержки 5 мс (при оборотах шпинделя около 5 000 об/м[1]), то сегодня пальма первенства принадлежит 3.5" дискам со средним временем доступа 5 мс и временем задержки 1 мс (при оборотах шпинделя 10 000 об/м). Здесь мы видим улучшение технических характеристик на величину около 100%. В тоже время, быстродействие процессоров увеличилось более чем на 2 000%. Во многом это стало возможно благодаря тому, что процессоры имеют прямые преимущества использования VLSI (сверхбольшой интеграции). Ее использование не только дает возможность увеличивать частоту, но и число компонент, которые могут быть интегрированы в чип, что дает возможность внедрять архитектурные преимущества, которые позволяют осуществлять параллельные вычисления.

[1] - Усредненные данные.

Сложившуюся ситуацию можно охарактеризовать как кризис ввода-вывода вторичной системы хранения данных.

Увеличиваем быстродействие

Невозможность значительного увеличения технологических параметров магнитных дисков влечет за собой необходимость поиска других путей, одним из которых является параллельная обработка.

Если расположить блок данных по N дискам некоторого массива и организовать это

размещение так, чтобы существовала возможность одновременного считывания информации, то этот блок можно будет считать в N раз быстрее, (без учёта времени формирования блока). Поскольку все данные передаются параллельно, это архитектурное решение называется *parallel-access array* (массив с параллельным доступом).

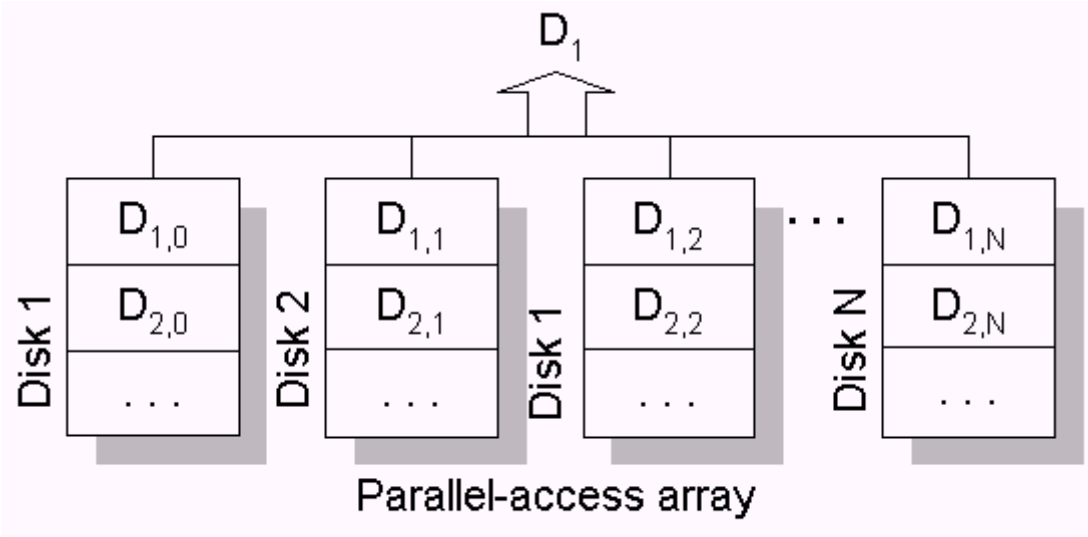

Рисунок 1.

Массивы с параллельным доступом обычно используются для приложений, требующих передачи данных большого размера.

Некоторые задачи, наоборот, характерны большим количеством малых запросов. К таким задачам относятся, например, задачи обработки баз данных. Располагая записи базы данных по дискам массива, можно распределить загрузку, независимо позиционируя диски. Такую архитектуру принято называть *independent-access array* (массив с независимым доступом).

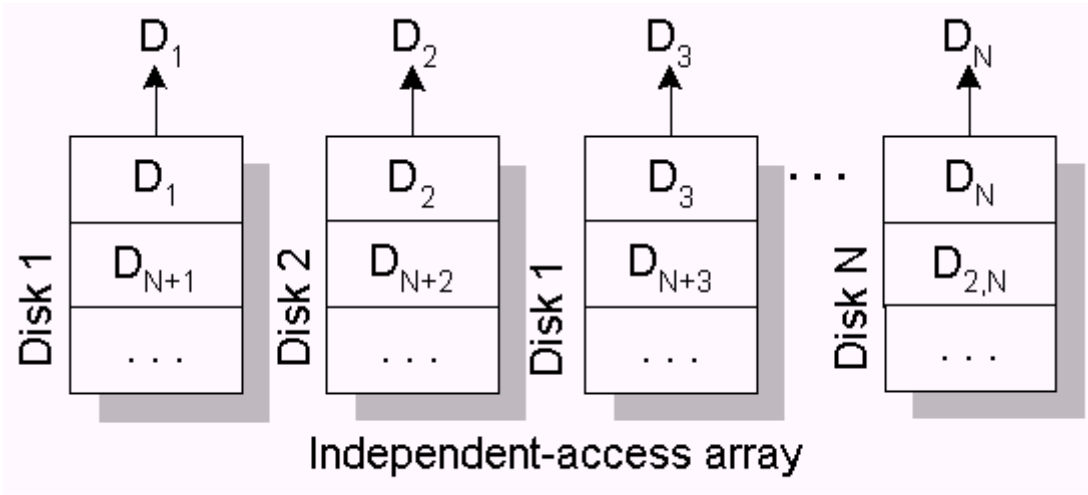

Рисунок 2.

Увеличиваем отказоустойчивость

К сожалению, при увеличении количества дисков в массиве, надежность всего массива уменьшается. При независимых отказах и экспоненциальном законе распределения наработки на отказ, MTTF всего массива (mean time to failure - среднее время безотказной работы) вычисляется по формуле $MTTF_{array} = MMTF_{hdd}/N_{hdd}$ ($MMTF_{hdd}$ - среднее время безотказной работы одного диска; NHDD -

количество дисков).

Таким образом, возникает необходимость повышения отказоустойчивости дисковых массивов. Для повышения отказоустойчивости массивов используют избыточное кодирование. Существует два основных типа кодирования, которые применяются в избыточных дисковых массивах - это дублирование и четность.

Дублирование, или зеркализация - наиболее часто используются в дисковых массивах. Простые зеркальные системы используют две копии данных, каждая копия размещается на отдельных дисках. Это схема достаточно проста и не требует дополнительных аппаратных затрат, но имеет один существенный недостаток - она использует 50% дискового пространства для хранения копии информации.

Второй способ реализации избыточных дисковых массивов - использование избыточного кодирования с помощью вычисления четности. Четность вычисляется как операция XOR всех символов в слове данных. Использование четности в избыточных дисковых массивах уменьшает накладные расходы до величины, исчисляемой формулой: $HP_{hdd}=1/N_{hdd}$ (HP_{hdd} - накладные расходы; N_{hdd} - количество дисков в массиве).

История и развитие RAID

Несмотря на то, что системы хранения данных, основанные на магнитных дисках, производятся уже 40 лет, массовое производство отказоустойчивых систем началось совсем недавно. Дисковые массивы с избыточностью данных, которые принято называть RAID (redundant arrays of inexpensive disks - избыточный массив недорогих дисков) были представлены исследователями (Петтерсон, Гибсон и Катц) из Калифорнийского университета в Беркли в 1987 году. Но широкое распространение RAID системы получили только тогда, когда диски, которые подходят для использования в избыточных массивах стали доступны и достаточно производительны. Со времени представления официального доклада о RAID в 1988 году, исследования в сфере избыточных дисковых массивов начали бурно развиваться, в попытке обеспечить широкий спектр решений в сфере компромисса - цена-производительность-надежность.

С аббревиатурой RAID в свое время случился казус. Дело в том, что недорогими дисками во время написания статьи назывались все диски, которые использовались в ПК, в противовес дорогим дискам для мейнфрейм (универсальная ЭВМ). Но для использования в массивах RAID пришлось использовать достаточно дорогостоящую аппаратуру по сравнению с другой комплектовкой ПК, поэтому RAID начали расшифровывать как redundant array of independent disks[2] - избыточный массив независимых дисков.

[2] - Определение RAID Advisory Board

RAID 0 был представлен индустрией как определение не отказоустойчивого дискового массива. В Беркли RAID 1 был определен как зеркальный дисковый массив. RAID 2 зарезервирован для массивов, которые применяют код Хемминга. Уровни RAID 3, 4, 5 используют четность для защиты данных от одиночных неисправностей. Именно эти уровни, включительно по 5-й были представлены в Беркли, и эта систематика RAID была принята как стандарт де-факто.

Для стандартизации продуктов RAID в 1992 году был организован промышленный консорциум - RAID Advisory Board. Подробно о работе консорциума можно узнать на сайте: www.raidadvisory.org.

Уровни RAID 3,4,5 достаточно популярны, имеют хороший коэффициент использования дискового пространства, но у них есть один существенный

недостаток - они устойчивы только к одиночным неисправностям. Особенно это актуально при использовании большого количества дисков, когда вероятность одновременного простоя более чем одного устройства увеличивается. Кроме того, для них характерно длительное восстановление, что также накладывает некоторые ограничения для их использования.

На сегодняшний день разработано достаточно большое количество архитектур, которые обеспечивают работоспособность массива при одновременном отказе любых двух дисков без потери данных. Среди всего множества стоит отметить two-dimensional parity (двухпространственная четность) и EVENODD, которые для кодирования используют четность, и RAID 6, в котором используется кодирование Reed-Solomon.

В схеме использующей двухпространственную четность, каждый блок данных участвует в построении двух независимых кодовых слов. Таким образом, если из строя выходит второй диск в том же кодовом слове, для реконструкции данных используется другое кодовое слово.

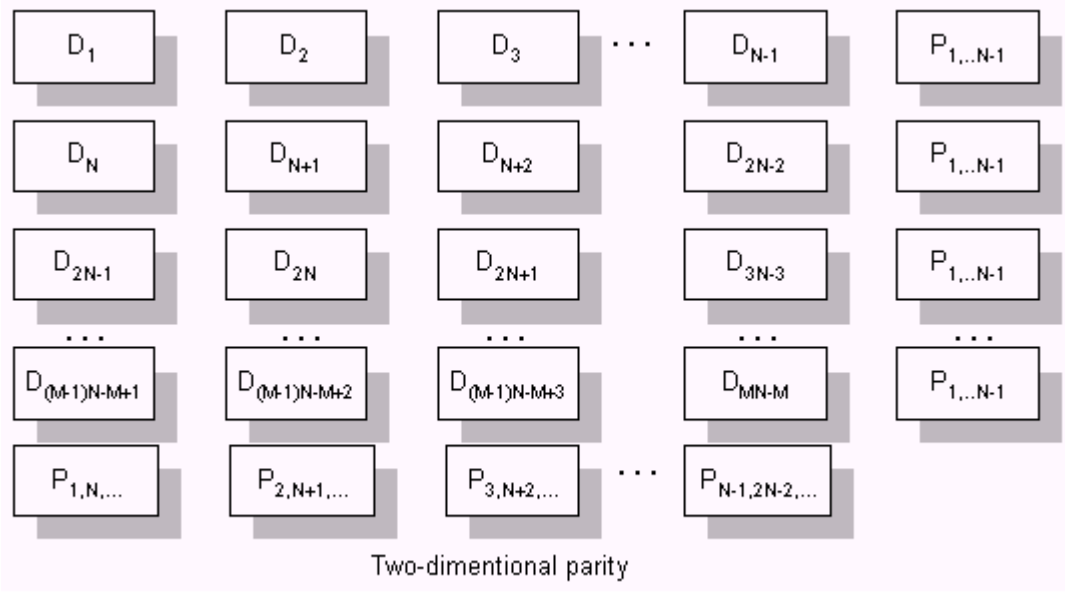

Рисунок 3. (На рисунке изображены диски).

Минимальная избыточность в таком массиве достигается при равном количестве столбцов и строчек. И равна: 2 x Square (N_{Disk}) (в "квадрат").

Если же двухпространственный массив не будет организован в "квадрат", то при реализации вышеуказанной схемы избыточность будет выше.

Архитектура EVENODD имеет похожую на двухпространственную четность схему отказоустойчивости, но другое размещение информационных блоков, которое гарантирует минимальное избыточное использование емкостей. Так же как и в двухпространственной четности каждый блок данных участвует в построении двух независимый кодовых слов, но слова размещены таким образом, что коэффициент избыточности постоянен (в отличие от предыдущей схемы) и равен: 2 x Square (N_{Disk}).

Используя два символа для проверки, четность и недвоичные коды, слово данных может быть сконструировано таким образом, чтобы обеспечить отказоустойчивость при возникновении двойной неисправности. Такая схема известна как RAID 6. Недвоичный код, построенный на основе Reed-Solomon кодирования, обычно вычисляется с использованием таблиц или как итерационный процесс с использованием линейных регистров с обратной связью, а это - относительно сложная операция, требующая специализированных аппаратных средств.

сложная операция, требующая специализированных аппаратных средств.

Учитывая то, что применение классических вариантов RAID, реализующих для многих приложений достаточную отказоустойчивость, имеет часто недопустимо низкое быстродействие, исследователи время от времени реализуют различные ходы, которые помогают увеличить быстродействие RAID систем.

В 1996 г. Саведж и Вилкс предложили AFRAID - часто избыточный массив независимых дисков (A Frequently Redundant Array of Independent Disks). Эта архитектура в некоторой степени приносит отказоустойчивость в жертву быстродействию. Делая попытку компенсировать проблему малой записи (small-write problem), характерную для массивов RAID 5-го уровня, разрешается оставлять стрипинг без вычисления четности на некоторый период времени. Если диск, предназначенный для записи четности, занят, то ее запись откладывается. Теоретически доказано, что 25% уменьшение отказоустойчивости может увеличить быстродействие на 97%. AFRAID фактически изменяет модель отказов массивов устойчивых к одиночным неисправностям, поскольку кодовое слово, которое не имеет обновленной четности, восприимчиво к отказам дисков.

Вместо того чтобы приносить в жертву отказоустойчивость, можно использовать такие традиционные способы увеличения быстродействия, как кэширование. Учитывая то, что дисковый трафик имеет пульсирующий характер, можно использовать кеш память с обратной записью (writeback cache) для хранения данных в момент, когда диски заняты. И если кеш-память будет выполнена в виде энергонезависимой памяти, тогда, в случае исчезновения питания, данные будут сохранены. Кроме того, отложенные дисковые операции, дают возможность объединить в произвольном порядке малые блоки для выполнения более эффективных дисковых операций.

Существует также множество архитектур, которые, принося в жертву объем, увеличивают быстродействие. Среди них - отложенная модификация на log диск и разнообразные схемы модификации логического размещение данных в физическое, которые позволяют распределять операции в массиве более эффективно.

Один из вариантов - *parity logging* (регистрация четности), который предполагает решение проблемы малой записи (small-write problem) и более эффективного использования дисков. Регистрация четности предполагает отложение изменения четности в RAID 5, записывая ее в FIFO log (журнал регистраций типа FIFO), который размещен частично в памяти контроллера и частично на диске. Учитывая то, что доступ к полному треку в среднем в 10 раз более эффективен, чем доступ к сектору, с помощью регистрации четности собираются большие количества данных модифицированной четности, которые потом все вместе записываются на диск, предназначенный для хранения четности по всему треку.

Архитектура *floating data and parity* (плавающие данные и четность), которая разрешает перераспределить физическое размещение дисковых блоков. Свободные сектора размещаются на каждом цилиндре для уменьшения *rotational latency* (задержки вращения), данные и четность размещаются на этих свободных местах. Для того, чтобы обеспечить работоспособность при исчезновении питания, карту четности и данных нужно сохранять в энергонезависимой памяти. Если потерять карту размещения все данные в массиве будут потеряны.

Virtual stripping - представляет собой архитектуру floating data and parity с использованием writeback cache. Естественно реализуя положительные стороны обеих.

Кроме того, существуют и другие способы повышения быстродействия, например распределение RAID операций. В свое время фирма Seagate встроила поддержку RAID операций в свои диски с интерфейсом Fibre Chanel и SCSI. Что дало возможность уменьшить трафик между центральным контроллером и дисками в

массиве для систем RAID 5. Это было кардинальным новшеством в сфере реализаций RAID, но технология не получила путевки в жизнь, так как некоторые особенности Fibre Chanel и SCSI стандартов ослабляют модель отказов для дисковых массивов.

Для того же RAID 5 была представлена архитектура TickerTAIP. Выглядит она следующим образом - центральный механизм управления originator node (узел-инициатор) получает запросы пользователя, выбирает алгоритм обработки и затем передает работу с диском и четность worker node (рабочий узел). Каждый рабочий узел обрабатывает некоторое подмножество дисков в массиве. Как и в модели фирмы Seagate, рабочие узлы передают данные между собой без участия узла-инициатора. В случае отказа рабочего узла, диски, которые он обслуживал, становятся недоступными. Но если кодовое слово построено так, что каждый его символ обрабатывается отдельным рабочим узлом, то схема отказоустойчивости повторяет RAID 5. Для предупреждения отказов узла-инициатора он дублируется, таким образом, мы получаем архитектуру, устойчивую к отказам любого ее узла. При всех своих положительных чертах эта архитектура страдает от проблемы "ошибки записи" ("write hole"). Что подразумевает возникновение ошибки при одновременном изменении кодового слова несколькими пользователями и отказа узла.

Следует также упомянуть достаточно популярный способ быстрого восстановления RAID - использование свободного диска (spare). При отказе одного из дисков массива, RAID может быть восстановлен с использованием свободного диска вместо вышедшего из строя. Основной особенностью такой реализации есть то, что система переходит в свое предыдущее (отказоустойчивое состояние без внешнего вмешательства). При использовании архитектуры распределения свободного диска (distributed sparing), логические блоки spare диска распределяются физически по всем дискам массива, снимая необходимость перестройки массива при отказе диска.

Для того чтобы избежать проблемы восстановления, характерной для классических уровней RAID, используется также архитектура, которая носит название *parity declustering* (распределение четности). Она предполагает размещение меньшего количества логических дисков с большим объемом на физические диски меньшего объема, но большего количества. При использовании этой технологии время реакции системы на запрос во время реконструкции улучшается более чем вдвое, а время реконструкции - значительно уменьшается.

Архитектура основных уровней RAID

Теперь давайте рассмотрим архитектуру основных уровней (basic levels) RAID более детально. Перед рассмотрением примем некоторые допущения. Для демонстрации принципов построения RAID систем рассмотрим набор из N дисков (для упрощения N будем считать четным числом), каждый из которых состоит из M блоков.

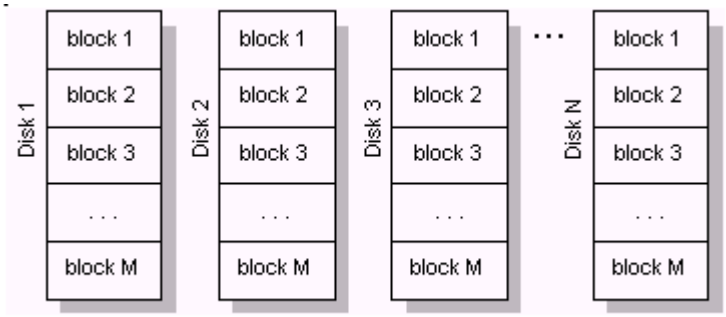

Рисунок 4.

Данные будем обозначать - $D_{m,n}$, где m - число блоков данных, n - число подблоков, на которые разбивается блок данных D.

Диски могут подключаться как к одному, так и к нескольким каналам передачи данных. Использование большего количества каналов увеличивает пропускную способность системы.

RAID 0. Дисковый массив без отказоустойчивости (Striped Disk Array without Fault Tolerance)

Представляет собой дисковый массив, в котором данные разбиваются на блоки, и каждый блок записываются (или же считывается) на отдельный диск. Таким образом, можно осуществлять несколько операций ввода-вывода одновременно.

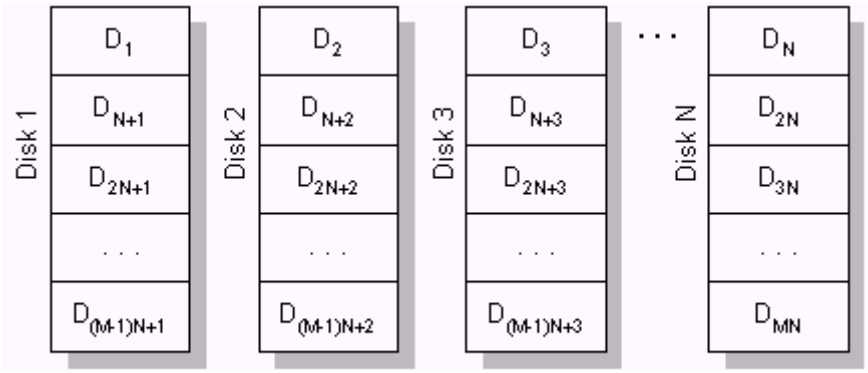

Рисунок 5.

Преимущества:

- наивысшая производительность для приложений требующих интенсивной обработки запросов ввода/вывода и данных большого объема;
- простота реализации;
- низкая стоимость на единицу объема.

Недостатки:

- не отказоустойчивое решение;
- отказ одного диска влечет за собой потерю всех данных массива.

RAID 1. Дисковый массив с дублированием или зеркалка (mirroring)

Зеркалирование - традиционный способ для повышения надежности дискового массива небольшого объема. В простейшем варианте используется два диска, на которые записывается одинаковая информация, и в случае отказа одного из них остается его дубль, который продолжает работать в прежнем режиме.

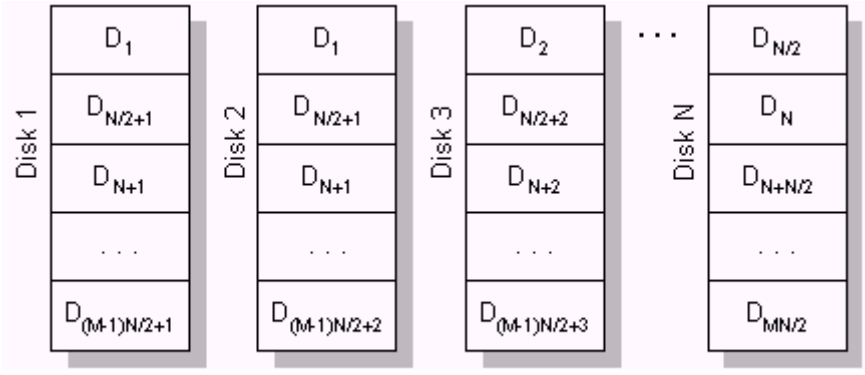

Рисунок 6.

Преимущества:

- простота реализации;
- простота восстановления массива в случае отказа (копирование);
- достаточно высокое быстродействие для приложений с большой интенсивностью запросов.

Недостатки:

- высокая стоимость на единицу объема - 100% избыточность;
- невысокая скорость передачи данных.

RAID 2. Отказоустойчивый дисковый массив с использованием кода Хемминга (Hamming Code ECC).

Избыточное кодирование, которое используется в RAID 2, носит название кода Хемминга. Код Хемминга позволяет исправлять одиночные и обнаруживать двойные неисправности. Сегодня активно используется в технологии кодирования данных в оперативной памяти типа ECC. И кодировании данных на магнитных дисках.

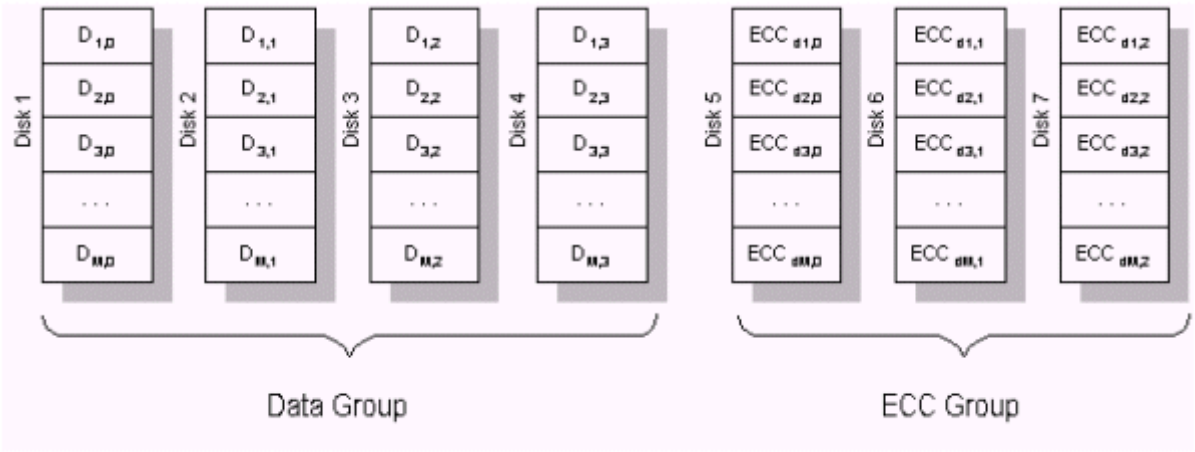

Рисунок 7.

В данном случае показан пример с фиксированным количеством дисков в связи с громоздкостью описания (слово данных состоит из 4-х бит, соответственно ECC код из 3-х).

Преимущества:

- быстрая коррекция ошибок ("на лету");
- очень высокая скорость передачи данных больших объемов;
- при увеличении количества дисков, накладные расходы уменьшаются;
- достаточно простая реализация.

Недостатки:

- высокая стоимость при малом количестве дисков;
- низкая скорость обработки запросов (не подходит для систем ориентированных на обработку транзакций).

RAID 3. Отказоустойчивый массив с параллельной передачей данных и четностью (Parallel Transfer Disks with Parity)

Данные разбиваются на подблоки на уровне байт и записываются одновременно на все диски массива кроме одного, который используется для четности

все диски массива кроме одного, который используется для четности. Использование RAID 3 решает проблему большой избыточности в RAID 2. Большинство контрольных дисков, используемых в RAID уровня 2, нужны для определения положения неисправного разряда. Но в этом нет нужды, так как большинство контроллеров в состоянии определить, когда диск отказал при помощи специальных сигналов, или дополнительного кодирования информации, записанной на диск и используемой для исправления случайных сбоев.

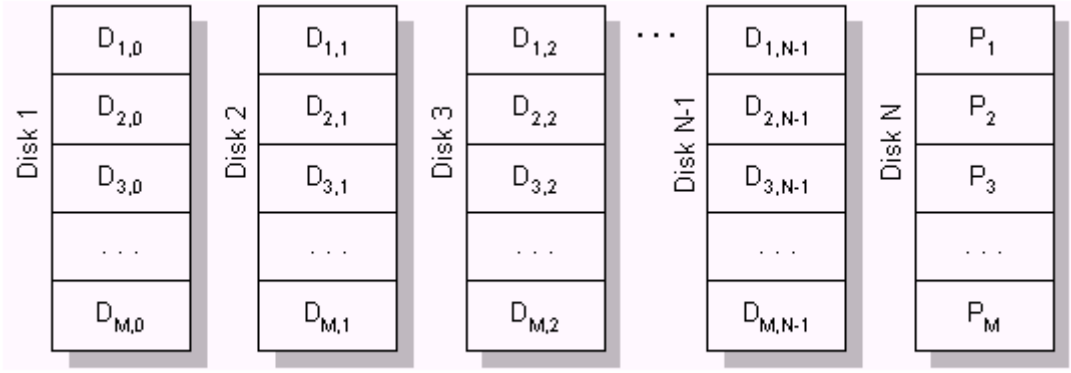

Рисунок 8.

Преимущества:

- очень высокая скорость передачи данных;
- отказ диска мало влияет на скорость работы массива;
- малые накладные расходы для реализации избыточности.

Недостатки:

- непростая реализация;
- низкая производительность при большой интенсивности запросов данных небольшого объема.

RAID 4. Отказоустойчивый массив независимых дисков с разделяемым диском четности (Independent Data disks with shared Parity disk)

Данные разбиваются на блочном уровне. Каждый блок данных записывается на отдельный диск и может быть прочитан отдельно. Четность для группы блоков генерируется при записи и проверяется при чтении. RAID уровня 4 повышает производительность передачи небольших объемов данных за счет параллелизма, давая возможность выполнять более одного обращения по вводу/выводу одновременно. Главное отличие между RAID 3 и 4 состоит в том, что в последнем, расслоение данных выполняется на уровне секторов, а не на уровне битов или байтов.

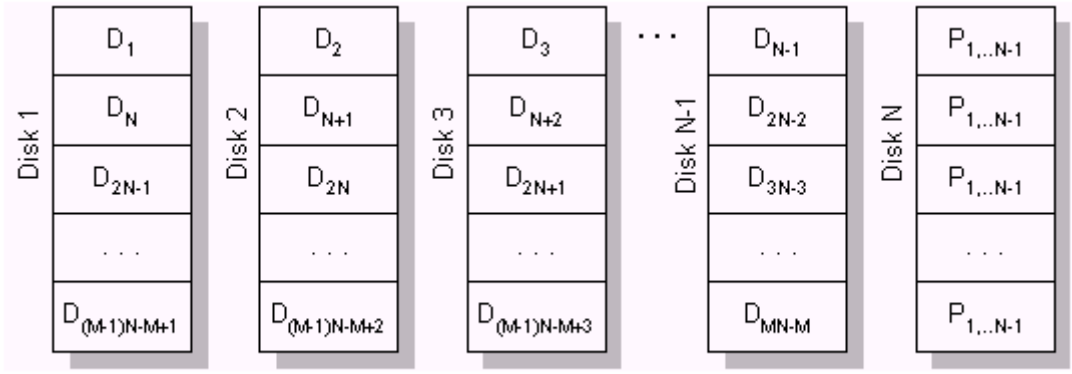

Рисунок 9.

Преимущества:

- очень высокая скорость чтения данных больших объемов;
- высокая производительность при большой интенсивности запросов чтения данных;
- малые накладные расходы для реализации избыточности.

Недостатки:

- достаточно сложная реализация;
- очень низкая производительность при записи данных;
- сложное восстановление данных;
- низкая скорость чтения данных малого объема при единичных запросах;
- асимметричность быстродействия относительно чтения и записи.

RAID 5. Отказоустойчивый массив независимых дисков с распределенной четностью (Independent Data disks with distributed parity blocks)

Этот уровень похож на RAID 4, но в отличие от предыдущего четность распределяется циклически по всем дискам массива. Это изменение позволяет увеличить производительность записи небольших объемов данных в многозадачных системах. Если операции записи спланировать должным образом, то, возможно, параллельно обрабатывать до N/2 блоков, где N - число дисков в группе.

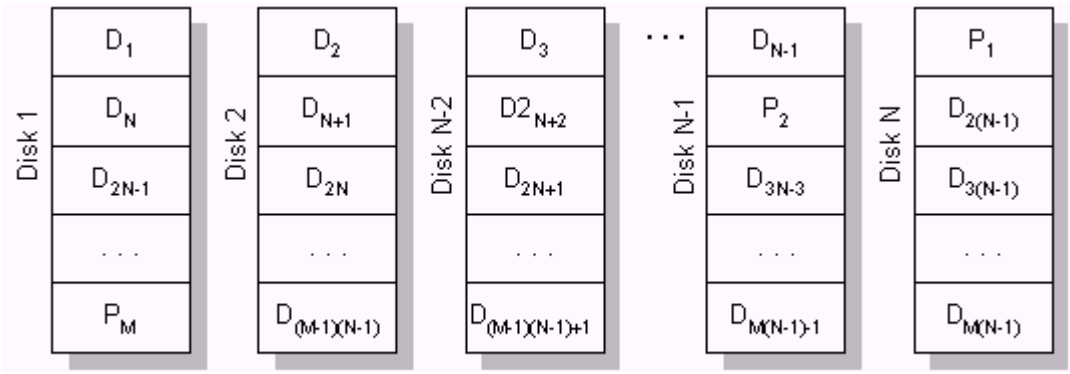

Рисунок 10.

Преимущества:

- высокая скорость записи данных;
- достаточно высокая скорость чтения данных;
- высокая производительность при большой интенсивности запросов чтения/записи данных;
- малые накладные расходы для реализации избыточности.

Недостатки:

- скорость чтения данных ниже, чем в RAID 4;
- низкая скорость чтения/записи данных малого объема при единичных запросах;
- достаточно сложная реализация;
- сложное восстановление данных.

RAID 6. Отказоустойчивый массив независимых дисков с двумя независимыми распределенными схемами четности (Independent Data disks with two independent distributed parity schemes)

Data disks with two independent distributed parity schemes)

Данные разбиваются на блочном уровне, аналогично RAID 5, но в дополнение к предыдущей архитектуре используется вторая схема для повышения отказоустойчивости. Эта архитектура является устойчивой к двойным отказам. Однако при выполнении логической записи реально происходит шесть обращений к диску, что сильно увеличивает время обработки одного запроса.

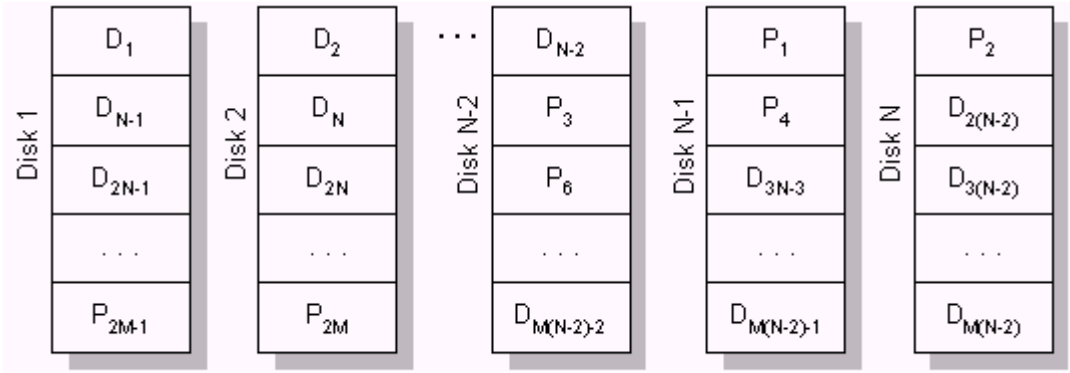

Рисунок 11.

Преимущества:

- высокая отказоустойчивость;
- достаточно высокая скорость обработки запросов;
- относительно малые накладные расходы для реализации избыточности.

Недостатки:

- очень сложная реализация;
- сложное восстановление данных;
- очень низкая скорость записи данных.

Современные RAID контроллеры позволяют комбинировать различные уровни RAID. Таким образом, можно реализовать системы, которые объединяют в себе достоинства различных уровней, а также системы с большим количеством дисков. Обычно это комбинация нулевого уровня (stripping) и какого либо отказоустойчивого уровня.

RAID 10. Отказоустойчивый массив с дублированием и параллельной обработкой

Эта архитектура являет собой массив типа RAID 0, сегментами которого являются массивы RAID 1. Он объединяет в себе очень высокую отказоустойчивость и производительность.

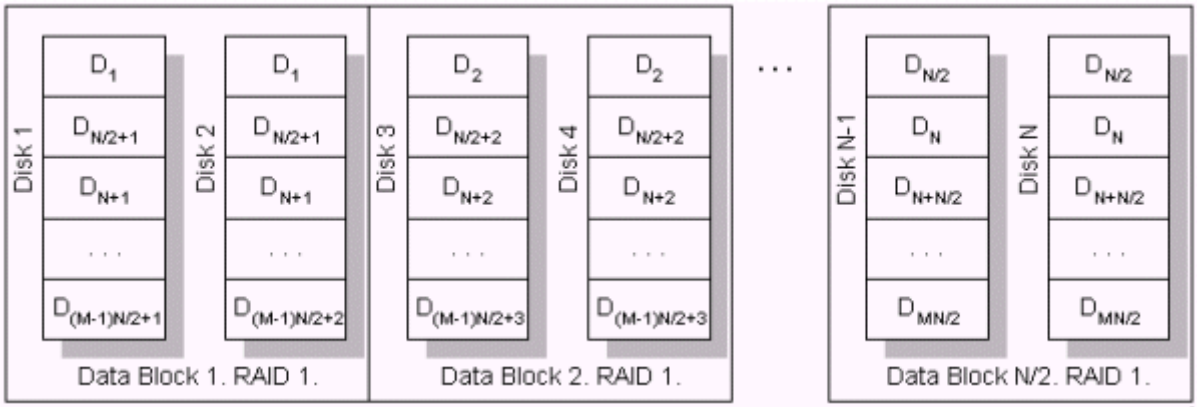

Рисунок 12.

Преимущества:

- высокая отказоустойчивость;
- высокая производительность.

Недостатки:

- очень высокая стоимость;
- ограниченное масштабирование.

RAID 30. Отказоустойчивый массив с параллельной передачей данных и повышенной производительностью.

Представляет собой массив типа RAID 0, сегментами которого являются массивы RAID 3. Он объединяет в себе отказоустойчивость и высокую производительность. Обычно используется для приложений требующих последовательной передачи данных больших объемов.

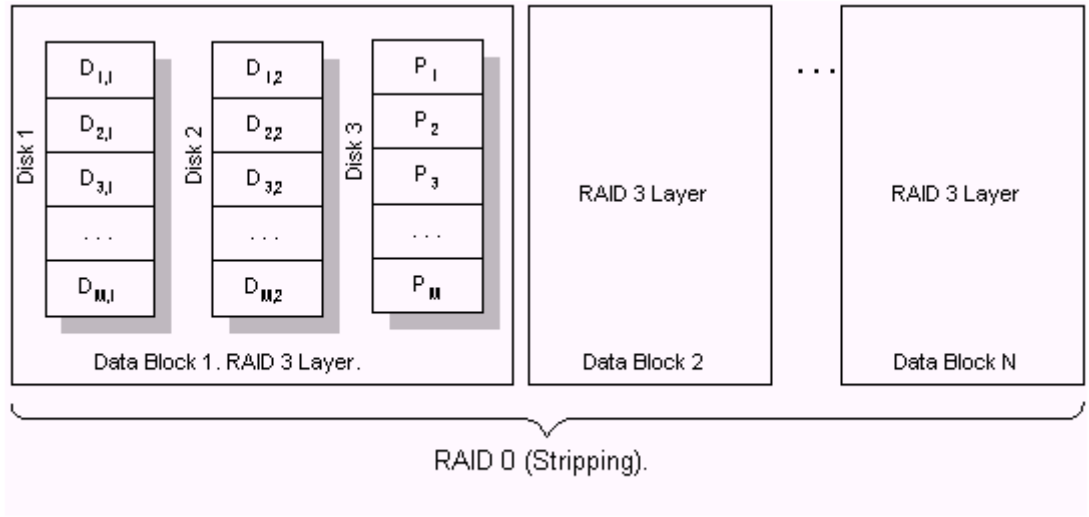

Рисунок 13.

Преимущества:

- высокая отказоустойчивость;
- высокая производительность.

Недостатки:

- высокая стоимость;
- ограниченное масштабирование.

RAID 50. Отказоустойчивый массив с распределенной четностью и повышенной производительностью

Являет собой массив типа RAID 0, сегментами которого являются массивы RAID 5. Он объединяет в себе отказоустойчивость и высокую производительность для

приложений с большой интенсивностью запросов и высокую скорость передачи данных.

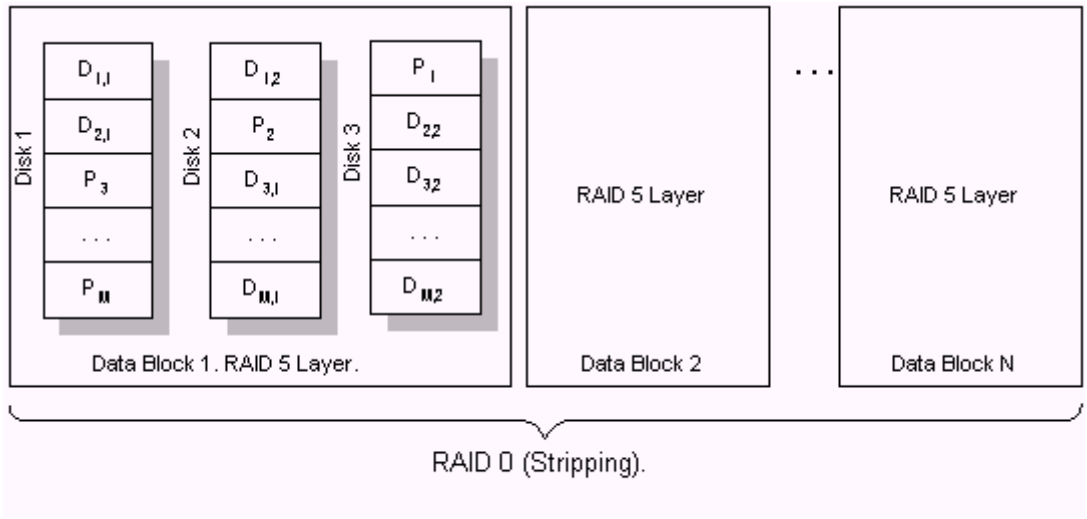

Рисунок 14.

Преимущества:

- высокая отказоустойчивость;
- высокая скорость передачи данных;
- высокая скорость обработки запросов.

Недостатки:

- высокая стоимость;
- ограниченное масштабирование.

RAID 7. Отказоустойчивый массив, оптимизированный для повышения производительности. (Optimized Asynchrony for High I/O Rates as well as High Data Transfer Rates). RAID 7® является зарегистрированной торговой маркой Storage Computer Corporation (SCC)

Для понимания архитектуры RAID 7 рассмотрим ее особенности:

1. Все запросы на передачу данных обрабатываются асинхронно и независимо.
2. Все операции чтения/записи кэшируются через высокоскоростную шину x-bus.
3. Диск четности может быть размещен на любом канале.
4. В микропроцессоре контроллера массива используется операционная система реального времени ориентированная на обработку процессов.
5. Система имеет хорошую масштабируемость: до 12-ти host-интерфейсов, и до 48-ми дисков.
6. Операционная система контролирует коммуникационные каналы.
7. Используются стандартные SCSI диски, шины, материнские платы и модули памяти.
8. Используется высокоскоростная шина X-bus для работы с внутренней кеш памятью.
9. Процедура генерации четности интегрирована в кеш.
10. Диски, присоединенные к системе, могут быть задекларированы как отдельно стоящие.
11. Для управления и мониторинга системы можно использовать SNMP агент.

Преимущества:

- высокая скорость передачи данных и высокая скорость обработки запросов (1.5 - 6 раз выше других стандартных уровней RAID);
- высокая масштабируемость хост интерфейсов;
- скорость записи данных увеличивается с увеличением количества дисков в массиве;
- для вычисления четности нет необходимости в дополнительной передаче данных.

Недостатки:

- собственность одного производителя;
- очень высокая стоимость на единицу объема;
- короткий гарантийный срок;
- не может обслуживаться пользователем;
- нужно использовать блок бесперебойного питания для предотвращения потери данных из кеш памяти.

Рассмотрим теперь стандартные уровни вместе для сравнения их характеристик. Сравнение производится в рамках архитектур, упомянутых в таблице.

RAID	Минимум дисков	Потребность в дисках	Отказо-устойчивость	Скорость передачи данных	Интенсивность обработки запросов	Практическое использование
0	2	N	< 1 диск	< RAID 3	очень высокая до N x 1 диск	Графика, видео
1	2	2N*	< RAID 6	R > 1 диск W = 1 диск	до 2 x 1 диск W = 1 диск	малые файл-серверы
2	7	2N<X<N+1	< RAID 1	~ RAID 3	Низкая	мейнфреймы
3	3	N+1	< RAID 1	< RAID 7	Низкая	Графика, видео
4	3	N+1	< RAID 1	R < RAID 3 W < RAID 5	R = RAID 0 W << 1 диск	файл-серверы
5	3	N+1	< RAID 1	R < RAID 4 W < RAID 3	R = RAID 0 W < 1 диск	серверы баз данных
6	4	N+2	самая высокая	низкая	R > 1 диск W < RAID 4	используется крайне редко
7	12	N+1	< RAID 1	самая высокая	самая высокая	разные типы приложений

Уточнения:

- * - рассматривается обычно используемый вариант;
- k - количество подсегментов;
- R - чтение;
- W - запись.

Некоторые аспекты реализации RAID систем

Рассмотрим три основных варианта реализации RAID систем:

- программная (software-based);
- аппаратная - шинно-ориентированная (bus-based);
- аппаратная - автономная подсистема (subsystem-based).

Нельзя однозначно сказать, что какая-либо реализация лучше, чем другая. Каждый вариант организации массива удовлетворяет тем или иным потребностям пользователя в зависимости от финансовых возможностей, количества пользователей и используемых приложений.

Каждая из вышеперечисленных реализаций базируется на исполнении программного кода. Отличаются они фактически тем, где этот код исполняется: в центральном процессоре компьютера (программная реализация) или в специализированном процессоре на RAID контроллере (аппаратная реализация).

Главное преимущество программной реализации - низкая стоимость. Но при этом у нее много недостатков: низкая производительность, загрузка дополнительной работой центрального процессора, увеличение шинного трафика. Программно обычно реализуют простые уровни RAID - 0 и 1, так как они не требуют значительных вычислений. Учитывая эти особенности, RAID системы с программной реализацией используются в серверах начального уровня.

Аппаратные реализации RAID соответственно стоят больше чем программные, так как используют дополнительную аппаратуру для выполнения операций ввода вывода. При этом они разгружают или освобождают центральный процессор и системную шину и соответственно позволяют увеличить быстродействие.

Шинно-ориентированные реализации представляют собой RAID контроллеры, которые используют скоростную шину компьютера, в который они устанавливаются (в последнее время обычно используется шина PCI). В свою очередь шинно-ориентированные реализации можно разделить на низкоуровневые и высокоуровневые. Первые обычно не имеют SCSI чипов и используют так называемый RAID порт на материнской плате со встроенным SCSI контроллером. При этом функции обработки кода RAID и операций ввода/вывода распределяются между процессором на RAID контроллере и чипами SCSI на материнской плате. Таким образом, центральный процессор освобождается от обработки дополнительного кода и уменьшается шинный трафик по сравнению с программным вариантом. Стоимость таких плат обычно небольшая, особенно если они ориентированы на системы RAID - 0 или 1 (есть также реализации RAID 3,5,10,30,50, но они дороже), благодаря чему они понемногу вытесняют программные реализации с рынка серверов начального уровня. Высокоуровневые контроллеры с шинной реализацией имеют несколько другую структуру, чем их младшие братья. Они берут на себя все функции, связанные с вводом/выводом и исполнением RAID кода. Кроме того, они не так зависимы от реализации материнской платы и, как правило, имеют больше возможностей (например, возможность подключения модуля для хранения информации в кеш в случае отказа материнской платы или исчезновения питания). Такие контроллеры обычно стоят дороже низкоуровневых и используются в серверах среднего и высокого уровня. Они, как правило, реализуют RAID уровней 0,1,3,5,10,30,50. Учитывая то, что шинно-ориентированные реализации подключаются прямо к внутренней PCI шине компьютера, они являются наиболее производительными среди рассматриваемых систем (при организации одно-хостовых систем). Максимальное быстродействие таких систем может достигать 132 Мбайт/с (32bit PCI) или же 264 Мбайт/с (64bit PCI) при частоте шины 33MHz.

Вместе с перечисленными преимуществами шинно-ориентированная архитектура имеет следующие недостатки:

- зависимость от операционной системы и платформы;
- ограниченная масштабируемость;
- ограниченные возможности по организации отказоустойчивых систем.

Всех этих недостатков можно избежать, используя автономные подсистемы. Эти

системы имеют полностью автономную внешнюю организацию и в принципе являют собой отдельный компьютер, который используется для организации систем хранения информации. Кроме того, в случае удачного развития технологии оптоволоконных каналов быстродействие автономных систем ни в чем не будет уступать шинно-ориентированным системам.

Обычно внешний контроллер ставится в отдельную стойку и в отличие от систем с шинной организацией может иметь большое количество каналов ввода/вывода, в том числе и хост-каналов, что дает возможность подключать к системе несколько хост-компьютеров и организовывать кластерные системы. В системах с автономным контроллером можно реализовать горячее резервирование контроллеров.

Одним из недостатков автономных систем остается их большая стоимость.

Учитывая вышесказанное, отметим, что автономные контроллеры обычно используются для реализации высокоемких хранилищ данных и кластерных систем.

Владимир Савяк, *(svv@ustar.kiev.ua)*
Опубликовано -- 12 марта 1999 года.

Комментарии? Поправки? Дополнения? niko@ixbt.com