разгон NTFS

крис касперски ака мыщъх no-email

> **дисковая подсистема — одно из самых "узких" мест компьютера, правильная настройка которой зачастую увеличивает производительность в несколько раз (в то время как разгон процессора в лучшем случае дает десятки процентов). мы не можем рассказать обо всех файловых системах, поэтому остановимся на NTFS как на самой популярной**

введение

Файловая система NTFS, автоматически устанавливаемая Windows 2000 и XP по умолчанию, привлекает все больше людей и через несколько лет на нее перейдут все или практически все. Прежде, чем погружаться в тонкости настройки NTFS, давайте обсудим: а стоит ли вообще на нее переходить? Ведь это не лучший выбор в плане производительности.

NTFS – это журналируемая файловая система, поддерживающая транзакции, значительно уменьшающие вероятность потери данных (можно даже вырубать питание в процессе дефргаментации), но вместе с тем вызывающие потери в производительности (иногда, очень значительные). На серверах и "серьезных" рабочих станиц сохранность данных превыше всего и другой альтернативы просто нет, но для домашнего компьютера выбор не столь очевиден и тут нужно крепко подумать.

К тому же NTFS требует намного больше памяти, имеет мерзкую проблему фрагментации $MFT (о ней мы еще поговорим), плохо справляется с каталогами, содержащими огромное количество файлов (хотя за счет индексации имен файлов на B*tree деревьях все должно быть наоборот) и вообще по сравнению с FAT32 конкретно тормозит.

А вот ее преимущества: поддержка разделов большого объема, возможность установки квот (т.е. выделенного объема дискового пространства) и прав разграничения доступа, прозрачное шифрование и упаковка отдельных файлов. Реально из этих пунктов в домашних компьютерах не используется ни один. Разделы большого объема? Это, конечно, хорошо, но по целому ряду соображений их лучше все-таки разбить.

Единственная причина, из-за которой мыщъх сидит под NTFS – это ее отказоустойчивость и надежность. Так что сворачиваем демагогию и переходим к изучению магических заклинаний, повышающих производительность

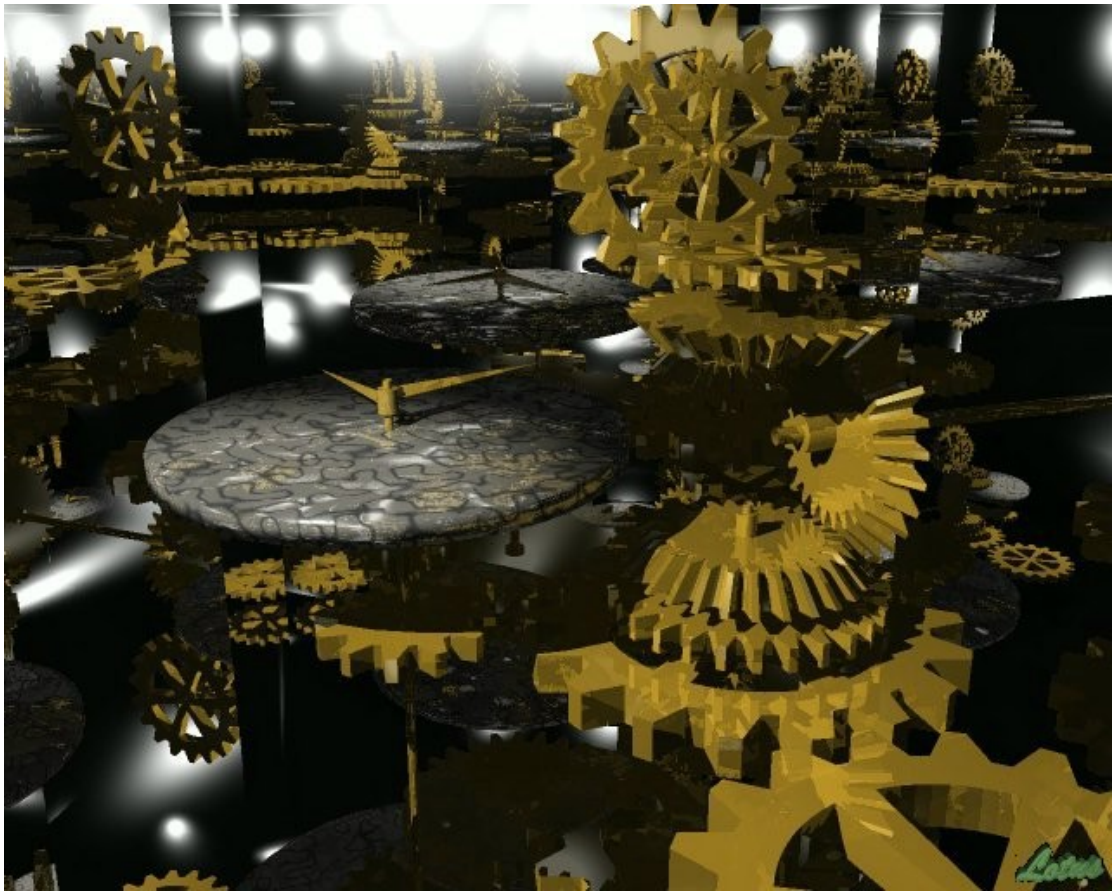

Рисунок 1 внутри NTFS

факторы, определяющие производительность

Вот пять факторов в наибольшей степени ответственных за производительность, перечисленных в порядке убывания собственной значимости:

- фрагментация;
- скорость диска;
- размер кластера;

Рисунок 2 мышиха дефрагментирует диск, слушая covenant

фрагментация

Знакомьтесь! Я — фрагментация! Я ужас, летящий на крыльях ночи, я — рокот ### стрекот магнитных головок, я — файл, размазанный по всей поверхности, я — жуткие тормоза. Я — ваш постоянный спутник и самый главный враг. Многие светлые головы пытались бороться со мной, оптимизируя алгоритм выделения свободного пространства файловой системой, но не многие в этом преуспели. NTFS всегда стремиться разместить весь файл в одном непрерывном куске целиком, да только где такой кусок найдешь? Вот и приходится дробить файл на части. Голова — в одном месте. Хвост — совсем в другом. Даже если файл не фрагментирован, но одновременно используемые файлы (скажем, исполняемый файл и файл данных) размешены в различных частях диска, магнитной головке приходится совершать большие телодвижения, а это — время.

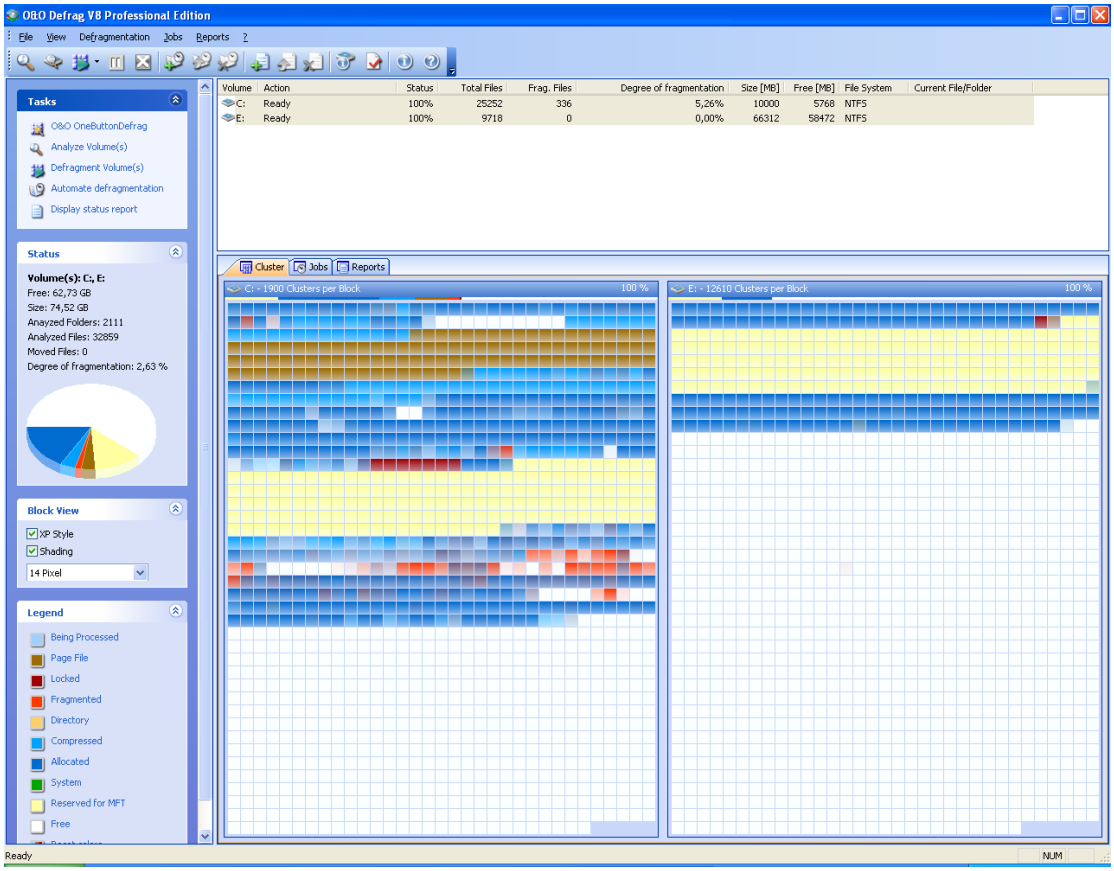

Рисунок 3 внешний вид дефрагментатора компании O&O (Professional Edition)

Предоставленная сама себе, фрагментация с течением времени неуклонно возрастает, замедляя работу компьютера без всяких видимых причин. Чтобы восстановить статус-кво, необходимо периодически (например, раз в месяц) запускать специальную утилиту — дефргаментатор. В штатную поставку Windows 2000 и XP входит что-то подобное, но это совсем не дефргаментатор, а жалкая пародия на него. Расследование показывает, что это усеченная версия одноименной утилиты, выпущенной компанией O&O, которую можно найти на сайте http://www.oo-software.com/ или в любом парнокопытном типа бобра.

Вот только некоторые, наиболее существенные различия:

- полная версия позволяет дефрагментировать открытые, системные, заблокированные файлы и MFT, штатный дефрагментатор — нет;
- полная версия поддерживает пять различных стратегий оптимизации, штатный дефрагментатор — только одну, самую худшую из них (дефрагментировать только наиболее фрагментированные файлы);
- полная версия умеет автоматически дефрагментировать диск при достижении заданного порога фрагментации, штатный дефрагментатор — нет;
- полная версия поддерживает фоновую дефраментацию по расписанию, штатный дефргаментатор — нет;
- полная версия может одновременно дефрагментировать все физические устройства, штатный дефрагментатор — нет;

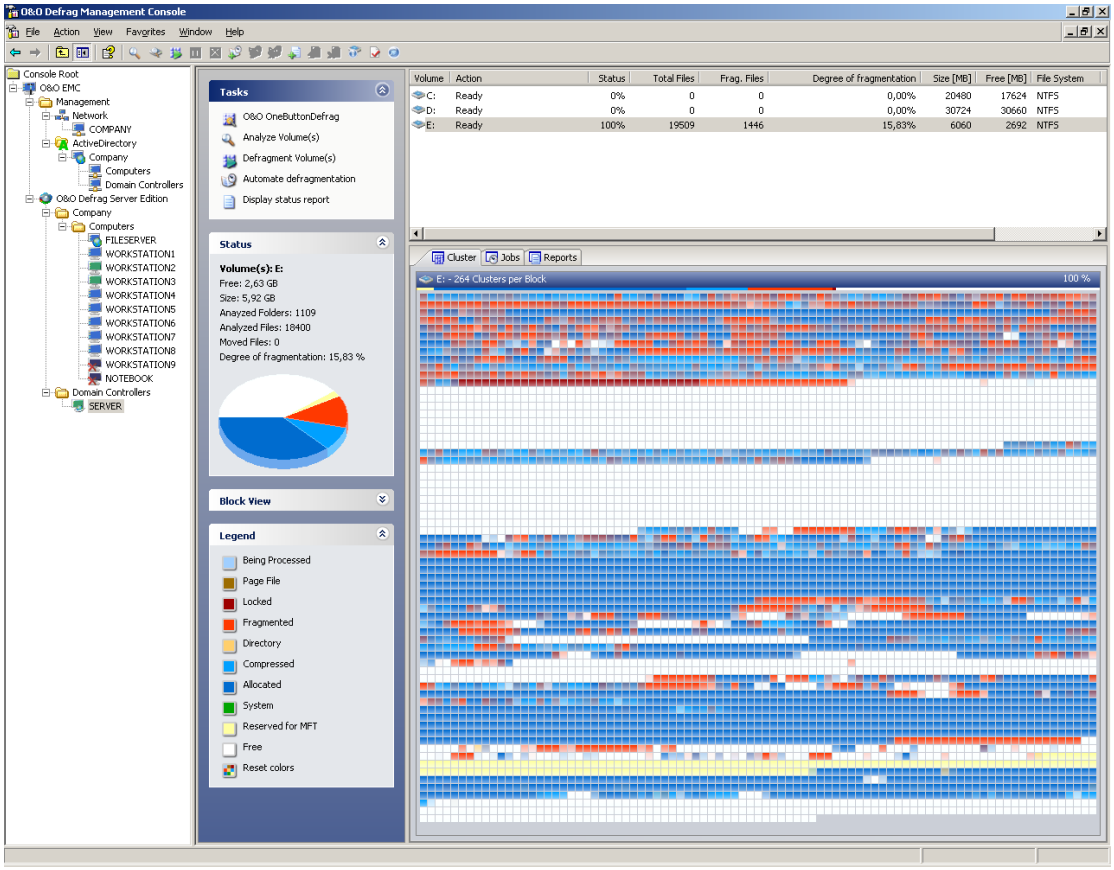

Рисунок 4 серверная редакция того же дефрагментатора

Мыщъх все может понять и простить (все-таки усеченная версия, доступная на халяву), но неумение дефрагментировтаь системные файлы (к которым в частности относятся реестр файл подкачки) и MFT превращает штатный дефрагментатор в игрушку. Сколько не дефрагментируй диск, производительность будет неуклонно снижаться, поскольку основные системные файлы остаются фрагментированными и ничего поделать с этим, увы, нельзя (отформатировать диск и переустановить Windows начисто не предлагать). Полная версия дефрагментатора запускается на самой ранней стадии загрузки системы и потому может дефрагментировать диск полностью!

Рисунок 5 внешний вид дефргаментатора от компании Symantec

>>> фрагментация MFT

$MTF (Master File Table – Главная Таблица Файлов) это служебный файл, хранящий важнейшие структуры данных, без которых NTFS не может функционировать. Он хранит имена, атрибуты и схему размещения всех файлов на диске (в том числе и самого себя, поскольку все структуры данных в NTFS представлены файлами).

Производительность NTFS во многом зависит от скорости доступа к $MFT-файлу. Обычно он располагается в начале раздела и резервирует 12,5% от объема не отформатированного раздела (не всего жесткого диска!), что предотвращает его фрагментацию. Однако, если свободное пространство заканчивается, NTFS делит остаток резерва напополам, одна половина остается за $MFT, другая — отдается на растерзание пользовательским файлам. Этот процесс может происходить многократно, до тех пор пока весь резерв не будет исчерпан целиком. Что происходит с $MFT, когда ему некуда дальше расти? (А ведь он растет!) Система находит свободный регион за пределами зарезервированной зоны и размешает продолжение $MFT там. То есть, $MFT не обязательно должен быть сосредоточен в одном месте и сохраняет свою работоспособность даже если его размазать по всему диску. Естественно, за экономию дискового пространства приходится расплачиваться скоростью, точнее полным отсутствуем таковой. Самое неприятное, что сокращение зарезервированной области обратного хода не имеет и она никогда не восстанавливается назад. Если хотя бы один раз заполнить диск более чем на 100 – 12,5% == 87,5% мы получим пожизненные тормоза!

Штатный дефрагментатор не умеет дефрагментировать $MFT, а это значит, что мы обречены на деградацию и неуклонное снижение производительности. Полная версия дефрагментирует $MFT, но "подсаживаться" на нее никому не в радость. Лучше надергать травы и сварить молока.

Рисунок 6 не всем нравится NTFS

Существует "магический" способ, позволяющий настроить размер зарезервированной области, выделяемой под $MFT (в разных источниках он называется и как MFT Breathing Room, и как MFT's buffer zone — устоявшегося термина нет), однако, он пригоден только для вновь создаваемых/форматируемых NTFS-томов, и не воздействует на уже существующие.

Запустите Редактор Реестр и найдите ветвь HKLM\SYSTEM\CurrentControlSet\Control\FileSystem и создайте там раздел NtfsMftZoneReservation типа REG_DWORD, если только она уже не была кем-то создана ранее. Теперь выберите значение по своему вкусу: 1 — резервирует 12,5% дискового пространства, 2 — 25%, 3 — 37,5%, 4 — 50%. Обратите внимание, что это именно резерв, а отнюдь не предельный размер $MFT, как утверждают некоторые левые твикеры. То есть это то пространство, которое будет выделено пользовательским файлам в последнюю очередь, но после того как это случиться, начнется необратимая фрагментация $MFT и тормоза. Сам же $MFT может расти сколько угодно.

Оценить приблизительный размер $MFT можно так: одна файловая запись (т. е. структура данных типа FILE Record) по умолчанию занимает 1 Кб. В грубом приближении, для каждого файла раздела создается одна FILE Record, однако для описания схема размещения сильно фрагментированных файлов может потребоваться несколько FILE Record. Тем не менее, в общем случае, размер $MFT в килобайтах равен количеству имеющихся файлов. На самом деле, это очень приближенная оценка, поскольку при удалении файлов немедленного уничтожения соответствующих им FILE Record не происходит и при создании новых файлов $MFT продолжает расти, вместо того, чтобы использовать уже освобожденные FILE Record'ы. Разумеется, система использует их, но… не сразу, а со временем. Конкретная стратегия нигде не описана и мыщъх подозревает, что она меняется от версии к версии, поэтому вывести точную формулу не получается, да она и не нужна. Возьмем раздел размером 1 ГГбайт. По умолчанию для $MFT файла будет выделено 128 Мбайт, что хватит для описания ~130.000 файлов. Да куда нам столько!

(Вообще-то, при желании возможно изменить размер зарезервированной области и без форматирования диска, и не только для $MFT, но и для любого файла, например, файла подкачки, реестра и т.д., однако, это довольно сложная операция, осуществляемая в суровых условиях дискового редактора, одно неверное движение которого способно угробить весь диск целиком, поэтому здесь эта тема не рассматривается, а всех любопытствующих мы отсылаем к

"технике восстановления данных" крис касперси, где это подробно описано, а сами продолжаем повествование).

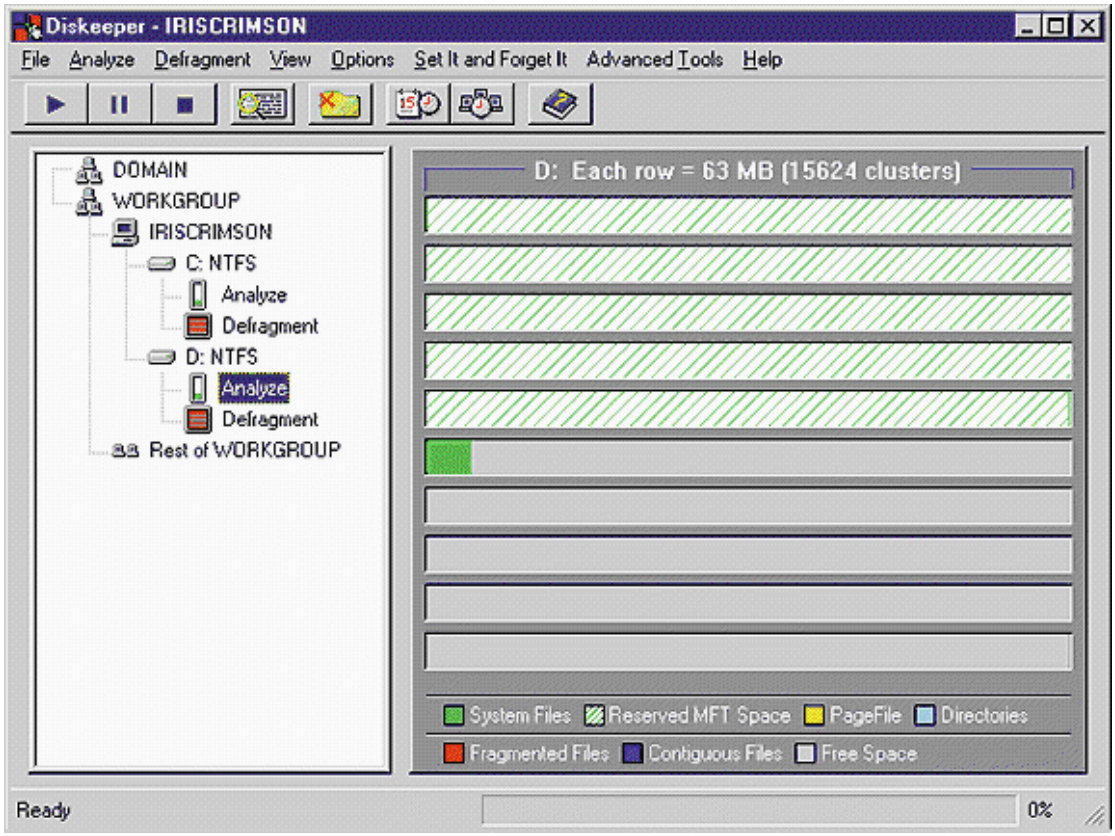

Рисунок 7 утилита Diskeeper из набора Microsoft Windows Resource Kit позволяет наблюдать за $MFT-файлов (выделен зеленым цветом)

>>> из FAT32 в NTFS

При создании NTFS раздела с "нуля" или его переформатировании, $MFT всегда размещается в начале раздела (где скорость доступа наиболее велика) и ему выделяется непрерывная область зарезервированного пространства. При обновлении существующего раздела типа FAT32 в NTFS все происходит иначе. Утилиты конвертации размещают $MFT там, где это получается, выискивая свободное пространство и размазывания служебную информацию по всему разделу. Как следствие — $MFT становится фрагментированным еще от рождения, причем очень сильно фрагментированным, причем первыми фрагментируются структуры, содержащие системные файлы и файлы, установленных приложений. Вот тебе бабушка и производительность на юрьев день.

Существует три пути выхода из ситуации: а) не обновлять FAT32 до NTFS, оставив все как есть (поверьте, и без NTFS все неплохо работает!); б) перенести все файлы в другое место и переформатировать раздел командой format с ключом /FS:NTFS; в) обновить FAT32 до NTFS стандартным путем и тут же запустить полную версию дефрагментатора O&O. Последний способ — наименее предпочтительный.

скорость диска

Скорость жесткого диска — величина постоянная и кажется, что никакой магией ее не поднять. На самом деле это заблуждение. Диск — источник великой производительности, особенно, если над ним поколдовать.

Начнем с извечной проблемы — бить или не бить? Мы будем рассуждать так: для достижения наивысшей производительности мы должны сократить перемещения магнитной головки, поскольку на операции позиционирования современные харды тратят намного больше времени, чем на последовательный доступ. Если служебная информация (типа $MFT) расположена в начале диска, а нужные нам файлы в конце, то головка будет метаться туда-сюда

и пройдет целая вечность, прежде чем она что-то прочтет. Когда профессор Эндрю Таненбаум проектировал учебную файловую систему для учебной операционной системы MINIX (из которой выросла LINUX), он это предусмотрел и учел, разделив раздел на несколько блоков, каждый из которых имеет "свою" служебную информацию и "свои" файлы. Как следствие, дистанция перемещения головки значительно сокращается, а скорость доступа — возрастает. В NTFS ничего подобного нет, но развив винчестер на несколько логических дисков, мы добьемся аналогичного эффекта. Мы так же сможем сгруппировать совместно используемые файлы на одном диске, чтобы время доступа к ним было минимальным.

Для еще большего увеличения производительности можно использовать два жестких диска, распараллелив операции чтения/записи. Варианты подключения тут самые различные. Диск, оставшийся от апгрейда, хорошо подходит для размещения файла подкачки (свойства системы → дополнительно → параметры быстродействия → виртуальная память → изменить), тогда, во-первых, он не будет фрагментироваться, во-вторых, головки не будут постоянно метаться по всей поверхности и, в-третьих, своп будет происходить одновременно с прочими операциями ввода/вывода, что увеличивает производительность в несколько раз. Еще лучше, разбить этот диск на два, разместив на втором временные файлы Интернета (Internet Explorer → свойства обозревателя → временные файлы → параметры → переместить) и системы (свойства системы → дополнительно → переменные среды → переменные TEMP и TMP).

Как альтернативный вариант, диски можно объединить в массив типа RAID0. Он может быть как программным, так аппаратный. В отличии от описанной выше схемы, он позволяет распараллеливать все дисковые операции, а не только обращение в файлу подкачки и временным файлам. На серверах и мощных рабочих станциях, занимающихся видеомонтажем или редредингом это дает существенный прирост производительности, но вот на домашнем компьютере практически не приносит никакой выгоды (ведь размер обрабатываемых файлов небольшой) и своп на отдельном диске рулит и выигрывает. Впрочем, современные материнские платы позволяют подключать намного больше двух IDE-устройств и потому можно легко организовать RAID0 + отдельный диск для свопа.

Считается, что совместно используемые диски ни в коем случае не должны висеть на одном шлейфе, иначе они будет работать последовательно. Это неверно. IDE устройства (точнее, не IDE, а ATA, но не будет углубляться в терминологиею) давным-давно поддерживают совместное использование шины и на правильных драйверах с нормальным чипсетом, мы практически полностью распараллеливаем операции ввода/вывода: ведь скорость передачи данных по шине намного выше скорости самого жесткого диска! Исключения составляют ситуации: захватил шишу и забыл отдать, но это уже глюк кривого железа. Исключение номер два: оптические приводы с винчестерами на одном шлейфе лучше не совмещать, поскольку у них всегда куча проблем с совместимостью.

Наконец, большинство винчестеров по умолчанию используют хитрую схему перемещения головки, учитывающую психоакустические особенности человеческого восприятия, то есть стараются шуметь как можно тише, издавая звуки, субъективно воспринимаемые как "приятные" или на худой конец "не раздражающие". Бесшумный компьютер это действительно здорово и мыщъх эту технологию одобряет всеми четырьмя ламами с хвостом в придачу, однако, любители разгона могут существенно увеличить скорость позиционирования, отключив "шмумоподавитель" и заставив головку двигаться по прямой. Обычно это делается специальными сервисными утилитами, которые можно нарыть на сайте производителя. Их достаточно запустить всего один раз, сохранив настройки в энергонезависимой памяти жесткого диска (все диски имеют такую).

Рисунок 8 это тоже NTFS

размер кластера

Большинство пользователей форматируют диски с размерами кластера по умолчанию и это правильно, поскольку система автоматически выбирает наилучшее. Чем меньше кластер, тем (потенциально) выше фрагментация, но меньше гранулярность, то есть потери дискового пространства. В NTFS никакой файл не может занять часть кластера, если он занимает хотя бы один байт, то весь кластер выделяется ему целиком (в файловых системах ReiserFS и FFS — использующихся в LINUX'e и BSD – это не так).

Если на диске храниться огромное количество мелких файлов, то сократив размер кластера, мы существенно увеличим эффективный объем (исключение составляют файлы, размер которых не превышает 1 Кбайт — они хранятся непосредственно в самом $MFT и размер кластера на них никак не влияет), однако за счет фрагментации производительность при этом упадет и нам придется намного чаще дефрагментировать свой диск.

Увеличение размера кластера хоть и увеличивает грануляцию (потерю дискового пространства), зато сдерживает рост фрагментации. В принципе, размер кластера можно выбирать любым, однако, если он будет больше 4 Кбайт, мы не сможем ни дефрагментировать раздел, ни использовать прозрачное сжатие/шифрование файлов, а это нехорошо.

размер раздела (Мбайт)	размер кластера (байт)	кол-во секторов в кластере
<= 512	512	1
513 – 1024	1024	2
1025 - 2048	2048	4
2049 >=	4096	8

Таблица 1 размер кластера, выбираемый операционной системой по умолчанию

Размер кластера задается при форматировании (в стандартном format'e за это отвечает ключ /A**:<clustersize>),** и потом не может быть изменен, поэтому выбирать его следует очень осторожно.

>>> твой друг reset

С точки зрения NTFS, директории – это обыкновенные файлы. Ну, может быть, и не совсем обыкновенные, но фрагментироваться они могут. Фот и фрагментируются. Причем,

штатный дефрагментатор их не дефрагментирует. Короче, как дальше жить… Перезагружаться надо почаще, вот что! Перед установкой нового приложения, создающего директории (или перед реорганизацией структуры каталогов "руками", ну типа в FAR'e или Проводнике), запустите штатный дефрагментатор и по окончании его работы обязательно перезагрузитесь. Не через reset, конечно, а завершив работу через меню "Пуск" как нормальные мыщъх'и

После этого файловая система готова к установке новых приложений. Весь фокус в том, что после перезагрузки NTFS стремится размещать директории вначале диска, а не черт знает где, как она это делает при нормальном развитии событий. Судя по всему, внутри системы существует специальный указатель, определяющий где будет расположена следующая директория, обнуляемый при перезагрузке. А быть может, и нет. Как бы там ни было, этот трюк действительно увеличивает скорость работы. Проверено мыщъх'ем. Факт!

Кстати говоря, глубоко вложенные директории лучше не создавать, т. к. это тормозит файловую систему. С другой стороны, директории с десятками тысячами файлов тормозят еще сильнее, поэтому здесь важно соблюдать баланс.

реестр черной магии

Реестр — это настоящий заповедный лес, населенных всякими существами, многие из которых очень полезны и годятся не только на мех, но еще и на оптимизацию. Вот несколько интересных ключей, влияющих на производительность NTFS:

отключение обновления времени последнего доступа

При каждом обращении к файлу, система автоматически обновляет время последнего доступа, что не только нарушает тайну конфиденциальности (начальник/администратор/жена сразу видит какие мы файлы открывали и когда), но еще и снижает производительность, правда незначительно. Но все-таки! Зачем нам это нужно?!

Запускаем Редактор Реестра, находим HKLM\SYSTEM\CurrentControlSet\Control \FileSystem, видим там параметр NtfsDisableLastAccessUpdate типа REG_DWORD со значением 0 (обновление включено). Меняем его на 1 (обновление отключено) и радуемся жизни.

отказ от коротких имен

Для каждого файла, NTFS автоматически генерирует длинное и короткое имя, сохраняя их в FILE Record. А зачем нам короткое имя, если его могут "видят" только старые 16-битные приложения, доставшиеся в наследство от Windows 3.x, о которых современное поколение, наверное, и не слышало.

Чтобы отключить короткие имена (они же имена формата 8.3 – восемь символов на имя, три — на расширение), берем HKLM\SYSTEM\CurrentControlSet\Control\FileSystem, переходим к параметру NtfsDisable8dot3NameCreation и меняем его значение с 0 (генерация коротких имен разрешена) на 1 (генерация коротких имен запрещена).

что в индексе твоем

Индексирование — это способ поиска такой. Грубо говоря, берем файл, выписываем все слова и складируем их в одну таблицу, просматривая которую мы сможем быстро сказать, присутствует ли искомая комбинация слов и где. Система поддерживает фоновое индексирование диска, ускоряющее поиск, только особого смысла в нем нет. На FAR (и другие популярные оболочки) оно не распространяется — поиск у них сугубо свой, да и Google Desktop Search работает намного быстрее. А ведь фоновое индексирование напрягает ЦП и вызывает дисковые тормоза. К счастью, его можно легко отключить. Для этого даже не нужно лезть в реестр. Достаточно в "Свойствах диска" снять галочку "Разрешить индексирование диска для быстрого поиска".

заключение

Работать на полностью вылизанной и дефрагментированной файловой системе — одно удовольствие. Все буквально летает! Правда, с таким трудом вычищенные Авгиевы Конюшни неизбежно пополняются свежим навозом, поэтому этот процесс приходиться повторять опять и опять. Но результат стоит того!

Рисунок 9 NTFS – оставайтесь с нами