Linux日志文件系统面面观 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/462/2021_2022_Linux_E6_97_A5_E5_BF_c103_462231.htm 文件系统是用来管理和组织保存在磁盘驱动器上的数据的系统软件，其实现了数据完整性的保证，也就是保证写入磁盘的数据和随后读出的内容的一致性。除了保存以文件方式存储的数据以外，一个文件系统同样存储和管理关于文件和文件系统自身的一些重要信息（例如：日期时间、属主、访问权限、文件大小和存储位置等等）。这些信息通常被称为元数据（metadata）。 由于为了避免磁盘访问瓶颈效应，一般文件系统大都以异步方式工作，因此如果磁盘操作被突然中断可能导致数据被丢失。例如如果出现这种情况：如果当你处理一个在linux的ext2文件系统上的文档，突然机器崩溃会出现什么情况？ 有这几种可能： 当你保存文件以后，系统崩溃。这是最好的情况，你不会丢失任何信息。只需要重新启动计算机然后继续工作。 在你保存文件之前系统崩溃。你会丢失你所有的工作内容，但是老版本的文档还会存在。 当正在将保存的文档写入磁盘时系统崩溃。这是最糟的情况：新版文件覆盖了旧版本的文件。这样磁盘上只剩下一个部分新部分旧的文件。如果文件是二进制文件那么就会出现不能打开文件的情况，因为其文件格式和应用所期待的不同。 在最后这种情况下，如果系统崩溃是发生在驱动器正在写入元数据时，那么情况可能更糟。这时候就是文件系统发生了损坏，你可能会丢失整个目录或者整个磁盘分区的数据。 linux标准文件系统（ext2fs）在重新启动时会通过调用文件扫描工具fsck试图恢复损坏的元数据信息。由

于ext2文件系统保存有冗余的关键元数据信息的备份，因此一般来说不大可能出现数据完全丢失。系统会计算出被损坏的数据的位置，然后或者是通过恢复冗余的元数据信息，或者是直接删除被损坏或是元数据信息损毁的文件。 很明显，要检测的文件系统越大，检测过程费时就越长。对于有几十个G大小的分区，可能会花费很长时间来进行检测。由于Linux开始用于大型服务器中越来越重要的应用，因此就越来越不能容忍长时间的当机时间。这就需要更复杂和精巧的文件系统来替代ext2. 因此就出现了日志式文件系统（journalling filesystems）来满足这样的需求。 什么是日志式文件系统 这里仅仅对日志式文件系统进行简单的说明。如果需要更深入的信息请参考文章日志式文件系统，或者是日志式文件系统介绍。 大多数现代文件系统都使用了来自于数据库系统中为了提高崩溃恢复能力而开发的日志技术。磁盘事务在被真正写入到磁盘的最终位置以前首先按照顺序方式写入磁盘中日志区（或是log区）的特定位置。 根据日志文件系统实现技术的不同，写入日志区的信息是不完全一样的。某些实现技术仅仅写文件系统元数据，而其他则会记录所有的写操作到日志中。 现在，如果崩溃发生在日志内容被写入之前发生，那么原始数据仍然在磁盘上，丢失的仅仅是最新的更新内容。如果当崩溃发生在真正的写操作时（也就是日志内容已经更新），日志文件系统的日志内容则会显示进行了哪些操作。因此当系统重启时，它能轻易根据日志内容，很快地恢复被破坏的更新。 在任何一种情况下，都会得到完整的数据，不会出现损坏的分区的情况。由于恢复过程根据日志进行，因此整个过程会非常快只需要几秒钟时间。 应该注

意的是使用日志文件系统并不意味着完全不需要使用文件扫描工具fsck了。随机发生的文件系统的硬件和软件错误是根据日志是无法恢复的，必须借助于fsck工具。 目前Linux环境下的日志文件系统 在下面的内容里将讨论三种日志文件系统：第一种是ext3，由Linux内核Stephen Tweedie开发。ext3是通过向ext2文件系统上添加日志功能来实现的，目前是redhat7.2的默认文件系统；Namesys开发的ReiserFs日志式文件系统，可以从www.namesys.com下载，目前Mandrake8.1采用该日志式文件系统。SGI在2001年三月发布了XFS日志式文件系统。可以在 oss.sgi.com/projects/xfs/下载。下面将对这三种日志文件系统采用不同的工具进行检测和性能测试。 安装ext3 关于ext3文件系统技术方面的问题请参考Dr. Stephen Tweedie的论文和访谈。ext3日志式文件系统直接来自于其祖先ext2文件系统。其具有完全向后兼容的关键特性，实际上其仅仅是在ext2日志式文件系统上添加了日志功能。其最大的缺点是没有现代文件系统所具有的能提高文件数据处理速度和解压的高性能。 ext3从 2.2.19开始是作为一个补丁方式存在的。如果希望对内核添加对ext3文件系统的支持，就需要使用补丁，可以从ftp.linux.org.uk/pub/linux/sct/fs/jfs 或ftp.kernel.org/pub/linux/kernel/people/sct/ext3得到补丁程序，一共需要如下文件：* ext3-0.0.7a.tar.bz2：内核补丁 * e2fsprogs-1.21-WIP-0601.tar.bz2 支持ext3的e2fsprogs程序套件 拷贝linux-2.2.19.tar.bz2和ext3-0.0.7a.tar.bz2到/usr/src目录下，进行解压： mv linux linux-old tar -Ixvf linux-2.2.19.tar.bz2 tar -Ixvf ext3-0.0.7a.tar.bz2 cd linux cat ⋯ ⋯/ext3-0.0.7a/linux-2.2.19.kdb.diff | patch -sp1 cat ⋯

…/ext3-0.0.7a/linux-2.2.19.ext3.diff | patch -sp1 首先对内核添加SGI的kdb内核调试器补丁，第二个是ext3文件系统补丁。下来就需要配置内核，对文件系统部分的"Enable Second extended fs development code"回答Yes.然后编译。 内核编译安装以后，需要安装e2fsprogs软件套件： tar -Ixvf e2fsprogs-1.21-WIP-0601.tar.bz2 cd e2fsprogs-1.21 。/configure make make check make install 下来要做的工作就是在分区上创建一个ext3文件系统，使用新内核重新启动，这时候你有两种选择创建新的日志文件系统或者对一个已有的ext2文件系统升级到ext3日志文件系统。 对于需要创建新ext3文件系统的情况下，只需要使用安装的e2fsprogs软件包中的mke2fs命令加-f参数就可以创建新的ext3文件系统： mke2fs -j /dev/xxx 这里/dev/xxx是希望创建ext3文件系统的新分区。-j参数表示创建ext3而不是ext2文件系统。可以使用参数"-Jsize="来指定希望的日志区大小（n单位为M）。 升级一个已有的ext2，使用tune2fs就可以了： tune2fs -j /dev/xxx 你可以对正在加载的文件系统和没有加载的文件系统进行升级操作。如果当前文件系统正在被加载，则文件。journal会在文件系统加载点的所在目录被创建。如果是升级一个当时没有加载的文件系统，则使用隐含的系统inode来记录日志，这时候文件系统的所有内容都会被保留不被破坏。 你可以使用下面的命令加载ext3文件系统： mount -t ext3 /dev/xxx /mount_dir 由于ext3实际上是带有日志功能的ext2文件系统，因此一个ext3文件系统可以以ext2的方式被加载。