

Linux Filesystem Performance
Comparison for OLTP
with Ext2, Ext3, Raw, and OCFS
on Direct-Attached Disks
using Oracle 9i Release 2

*An Oracle White Paper
January 2004*

Linux Filesystem Performance Comparison for OLTP

Executive Overview	3
Filesystems.....	4
Ext2 Filesystem	4
Ext3 Filesystem	5
Raw Devices	5
Oracle Clustered Filesystem (OCFS)	6
Benchmark	6
Configuration	6
Test Setup	7
Test Goals	7
Results	8
Conclusion.....	10
Appendix A – Datafile Layout	11
Appendix B – Init.ora Parameter File	12

Linux Filesystem Performance Comparison for OLTP

EXECUTIVE OVERVIEW

This paper describes the technical background and the performance results of a transaction processing workload that we ran on an Oracle 9i Release 2 database server to compare the performance of four Linux filesystems on direct-attached SCSI storage. We tested two traditional filesystems, ext2 and ext3; and we tested two special-purpose filesystems, raw device pseudo-files and the Oracle Cluster Filesystem (OCFS).

In general, filesystems store and manage structured information on disk partitions which vary in size depending on the configuration and needs of the applications. Filesystems are generally such a crucial part of computer systems that the reliability and performance of most applications greatly depend on the reliability and performance of the underlying filesystem.

However; unlike most applications the Oracle database server itself also implements many of the same functions that are implemented by a traditional filesystem including cache management, IO optimization and data layout. This overlap of filesystem and database functionality raises several questions: How does Oracle on Linux perform when the storage is provided by a typical filesystem? Can the two systems work together? How well can the database perform when it controls the storage more directly? These questions motivated the performance tests discussed in this paper.

The test results show that OCFS and raw devices performed similarly; that ext2 and ext3 performed similarly; and that as the workload scaled up, OCFS and raw device files yielded significantly greater performance than the ext2 and ext3 filesystems. The following sections discuss the details of the tests and the results.

FILESYSTEMS

This section briefly describes the filesystems we tested: ext2, ext3, raw, and OCFS.

Nearly all Linux filesystems, including ext2 and ext3, use the buffer cache layer in the Linux operating system kernel for disk reading or writing. The kernel not only caches data but also uses algorithms such as read-ahead (which consecutively reads extra data blocks into the cache in the hope that these will be the next data blocks requested by the application).

As it turns out, although the kernel buffer cache layer is beneficial in many applications it will usually decrease performance for a database application such as Oracle. Why? Primarily because the Oracle database itself already has a buffer cache, so the double-caching of data wastes system memory; furthermore the Oracle database has more knowledge of its client IO access patterns than the filesystem can have, so the cache buffer replacement strategies and other algorithms (such as read-ahead) are most likely to improve performance if handled in the database rather than the filesystem. A variety of other factors can also decrease database performance in a filesystem; for example some OS filesystem implementations develop resource bottlenecks when Oracle uses large numbers of files.

It is important to note that the files which most directly affect the performance of a database workload such as the OLTP benchmark discussed in this paper are the Oracle database's *data* files. In this paper the filesystems are compared solely for the performance impact on the chosen benchmark, which means that only the datafiles are stored in the filesystems for our test (see Appendix A). The following are some examples of other files used with Oracle that are *not* data files: executable binaries, message files, trace files, and shared libraries.

Ext2 Filesystem

Ext2, which is short for “second extended filesystem,” was the de facto standard on numerous Linux distributions for many years. Ext2 is a reliable and robust filesystem and provides a rich set of features including subdirectories, attributes, quotas, and locks.

One potential problem with ext2 and many other filesystems is that in case of an improper system shutdown such as a power failure, an ext2 filesystem cannot be

used until a filesystem consistency check (fsck) has been performed. The time taken for the consistency check is heavily dependent on the size of the filesystem: the bigger the filesystem, the longer it takes to finish the consistency check. With modern disk capacities reaching hundreds of Gigabytes (and doubling each year), it can take hours or even days to check large filesystems – and during this time the applications cannot run and the system is unavailable.

Ext3 Filesystem

Ext3 is now available on most Linux distributions. Ext3 is an enhancement of ext2 to implement algorithms for efficiently using a journal of all writes to the on-disk filesystem. The journal itself is also stored on the disk which enables ext3 to be reliable, and the IO to the journal is sequential (due to the physical layout of the journal's data blocks on the disk) which enables ext3 to perform well.

Ext3 has a number of advantages. A great advantage of ext3 is that when you have a large disk and need to recover from a system crash, the recovery process (filesystem consistency check) is much faster than on Ext2. By journaling changes (writes), ext3 can recover very quickly regardless of the size of the filesystem. Another advantage is that ext3 is compatible with Ext2 so converting filesystems between Ext2 and Ext3 is very easy and does not need a reboot or repartitioning of the disks.

An ext3 filesystem can be mounted in one of 3 modes:

- data=journal – this is the most reliable, but slowest of the 3 modes
- data=ordered – this is the default mode
- data=writeback – this is generally the fastest of the 3 modes

Our performance tests included these three modes of ext3 operation. However, no further filesystem tuning was done.

Raw Devices

Linux provides raw device access with a pseudo filesystem which presents a file-like interface to read and write actual disk partitions. Normally reads and writes to files go through the Linux filesystem buffer cache; however, raw device IO bypasses the buffer cache and directly deals with the underlying hardware devices. There is one raw device for one partition. For the 2.4 series Linux operating system kernel that was used for this performance comparison, the maximum number of raw devices that a system can have is fixed at a total of 255 raw devices, and the number of partitions that can be created on any disk is fixed at 14.

Oracle Clustered Filesystem (OCFS)

OCFS is Oracle's open-source filesystem available on various platforms. This is an extent-based (an extent is a variable contiguous space) filesystem which is currently intended for Oracle datafiles and Oracle RAC (real application clusters). In terms of the file interface it provides to applications, OCFS balances performance and manageability by providing functionality that is in-between the functionality provided by raw devices and typical filesystems. While retaining the performance of raw devices (as we see from the results of the benchmark in the next section), OCFS provides higher-order, more manageable file interfaces. In this respect, the OCFS service can be thought of as a filesystem-like interface to raw devices. At the same time, the cluster features of OCFS go well beyond the functionality of a typical filesystem. OCFS files can be shared across multiple computers, or nodes, on a network so that the files are simultaneously accessible by all the nodes, which is essential in RAC configurations. For example, sharing datafiles allows media recovery in case of failures, as all the datafiles (archive log files) are visible from the nodes that constitute the RAC cluster.

Beyond clustering features and basic file service, OCFS provides a number of manageability benefits (for example, resizing datafiles and partitions is easy) and comes with a set of tools to manage OCFS files.

For more information on OCFS, please visit <http://otn.oracle.com/linux>.

BENCHMARK

This test used an OLTP workload, generated by processes simulating users who connect to a database and perform transactions. The database simulated a real-world retail store chain with 1000 warehouses. By varying the number of users, we controlled the amount of work and the load on the CPU and the IO subsystem.

Configuration

Hardware

<i>Processor</i>	4 x Intel Pentium III 700 Mhz (Cache 2MB)
<i>Storage</i>	96 x 8 GB SCSI direct-attached disks (configured as 8 disks in RAID-0)
<i>Disk Controller</i>	MegaRAID v1.73
<i>Memory</i>	4 GB

Software

<i>Linux Distribution</i>	Red Hat Advanced Server 2.1
<i>Kernel</i>	2.4.9-e12smp
<i>OCFS</i>	1.0.8-4
<i>Oracle Database</i>	9i Release 2, version 9.2.0.3

Test Setup

The setup consisted of the above server with direct-attached storage. The database files were distributed evenly over 6 disks. The disk layout is mentioned in Appendix A. Appendix B contains the database parameter file used for the tests.

Test Goals

Evaluate the performance of ext2, ext3, raw, and OCFS in terms of the following parameters, which are representative of the database performance.

- **Transactions Per Second (TPS)** The amount of work done per second; this is a measure of throughput. The more transactions, the better the overall system performance.
- **Input/Output (IO)** The amount of IO done, in kilobytes per second; this is a measure of bandwidth.
- **CPU Utilization** The average CPU used during a test run given as a percentage where 100% = full CPU utilization; this is a measure of system processing load.

From past experience on this hardware, IO throughput was found to be optimum for benchmark simulations of 100 users and less, so for this paper the test plan was to test only up to 100 users. However; since the raw and OCFS results scaled linearly to 100 users, in retrospect a higher threshold could have been used.

RESULTS

TPS measures the rate at which work is done. The more transactions completed per second, the better the overall performance. Since the TPS value indicates the overall database performance, it is the chief metric we used to compare the filesystems.

The three modes of ext3 operation are journaled, ordered, and writeback. We denote them here by ext3j, ext3o, and ext3w, respectively.

Figure 1: TPS Graph

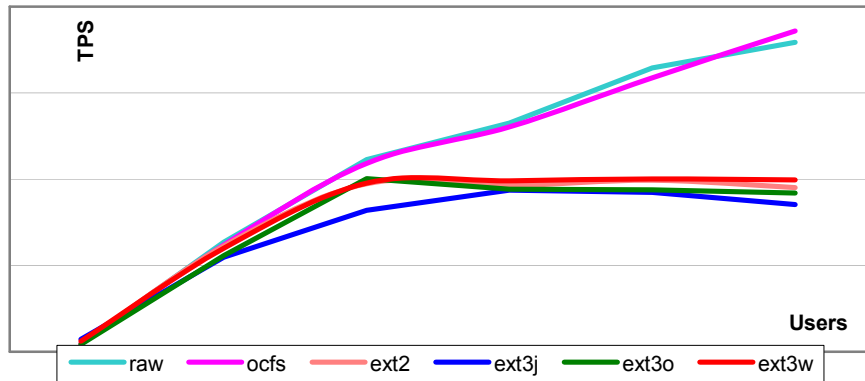
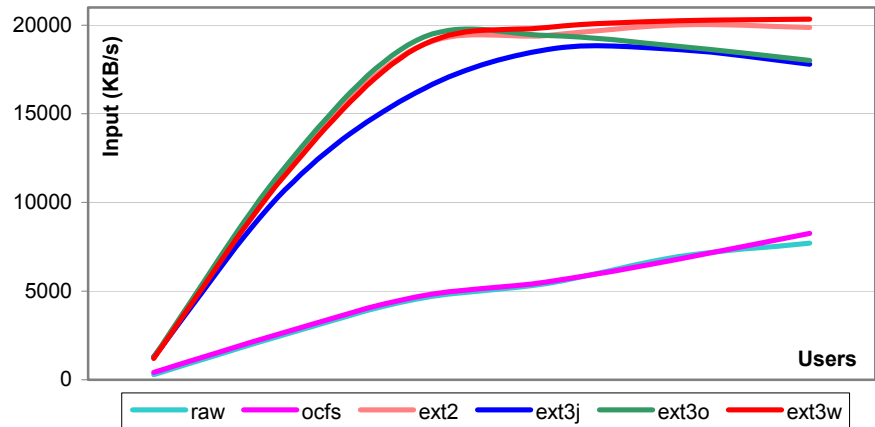


Figure 1 shows that when using raw or OCFS the transaction throughput increases linearly. Since raw and OCFS bypass the filesystem cache, there is more memory available for data. For Ext2/Ext3, there is a linear increase for some time after which the TPS graph stays level. The performance increase stops as the database cache gets filled up. To acquire free buffers the database writer needs to clear some space in its own cache, and the system performance becomes IO-bound because the database is busy writing these disk blocks. Thus we see more “free buffer waits” in the Oracle statspack reports.

The input bandwidth in kilobytes per second measures the amount of data read by the system from the disks. We used OS utilities to measure the rate of input,

Figure 2: Input Graph



As seen in Figure 2, the number of bytes read from disk is higher for Ext2/Ext3, than for raw and OCFS. This is attributable to filesystem read-ahead IO and the reduced memory available for the database due to the filesystem cache.

Since raw and OCFS bypass the filesystem cache, they do not have any read-ahead (Oracle itself may issue read-ahead requests when they will directly benefit database performance, but the raw and OCFS filesystems by themselves do not).

The output bandwidth in kilobytes per second measures the amount of data written to storage by the system. We used OS utilities to measure the rate of output,

Figure 3: Output Graph

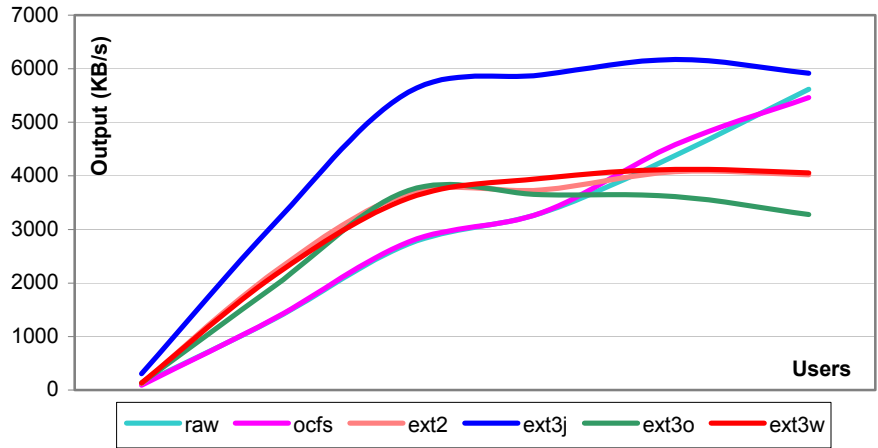


Figure 3, above, shows several effects. First, for ext2 and ext3 the number of bytes written is greater than for raw or OCFS until the workload is scaled to a high number of users. Why? Initially there are more Oracle buffer cache flushes for ext2 and ext3, but for high numbers of users the system does less total work than with raw and OCFS (see also Figure 1 above). Second, the output for ext2/ext3 eventually stays constant which is likely due to a bottleneck in the IO subsystem. Also, in journal mode ext3 writes more because it journals both data and metadata.

The percentage of full CPU utilization measures the processor load and also indirectly provides information about relative system IO load. We used OS utilities to measure CPU utilization.

Figure 4: CPU Utilization Graph

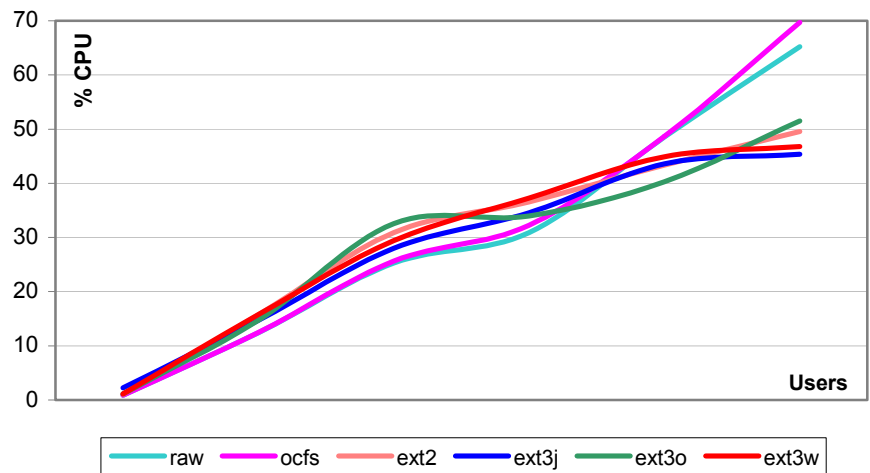


Figure 4 shows the CPU utilization for all the four filesystems being tested in this paper, namely ext2, ext3, raw, and OCFS. In all cases, we observe that the processor load increases with the number of users, as expected. For very high numbers of users the ext2 and ext3 filesystems do less processing because their IO rates are higher and the CPU is less busy while waiting for IO. It is important to note that there is also an opposite effect on the CPU activity: whenever the IO increases as it does for ext2 and ext3 under very high benchmark load, the Linux kernel activity such as queuing, swapping, and other resource management will take up more CPU time than under more normal conditions. However; the decrease in benchmark processing due to heavier IO load usually outweighs the increase in system CPU. Furthermore, this additional system CPU load actually subtracts from the total processing power available to do directly useful work (i.e. database transactions).

CONCLUSION

By design, this was purely a performance comparison (factors such as cost, manageability, and reliability were not considered) and used one database workload, albeit a typical one. We ran the same database benchmark using 4 different Linux filesystems for data storage and we measured transaction throughput, IO activity, and server CPU utilization. Transaction throughput was the primary metric of the benchmark. The results showed that raw and OCFS performance are nearly identical and they yield better overall performance for the OLTP database workload that we tested.

The ext2 and ext3 filesystems have a cache which decreases the database workload performance for increasing users/load on the system. For small number of users, ext2 or ext3 may suffice but for large number of users, the limits of the IO subsystem are reached sooner than with raw and OCFS storage.

The performance of the overall system depends on the type of application and for an Oracle database it usually depends greatly on the physical organization of data on disk. Traditional filesystems such as ext2 and ext3 may perform better for simple applications which benefit from the caches and algorithms of the filesystem layer. However; for Oracle database systems the on-disk data layout and access strategies are best left to the Oracle server and our test results demonstrate the performance advantages of storing Oracle data in either raw files or OCFS files.

APPENDIX A – DATAFILE LAYOUT

<u>Disk/Partition</u>	<u>Raw Device</u>	<u>Database File</u>	<u>Size (MB)</u>
/dev/sde7	/dev/raw/raw70	stok_0_0	7500
/dev/sde8	/dev/raw/raw77	icom_0_0	3500
/dev/sde9	/dev/raw/raw84	dcom_0_0	2500
/dev/sde10	/dev/raw/raw91	ordl_0_0	8032
/dev/sde11	/dev/raw/raw98	temp_0_0	3000
/dev/sde12	/dev/raw/raw105	cust_0_0	5000
/dev/sde13	/dev/raw/raw118	sp_0	1000
/dev/sdf7	/dev/raw/raw71	stok_0_1	7500
/dev/sdf8	/dev/raw/raw78	icom_0_1	3500
/dev/sdf9	/dev/raw/raw85	dcom_0_1	2500
/dev/sdf10	/dev/raw/raw92	ordl_0_1	8032
/dev/sdf11	/dev/raw/raw99	temp_0_1	3000
/dev/sdf12	/dev/raw/raw106	cust_0_1	5000
/dev/sdf13	/dev/raw/raw114	ctl1	1000
/dev/sdg8	/dev/raw/raw72	stok_0_2	7500
/dev/sdg9	/dev/raw/raw79	icom_0_2	3500
/dev/sdg10	/dev/raw/raw86	dcom_0_2	2500
/dev/sdg11	/dev/raw/raw93	ordl_0_2	8032
/dev/sdg12	/dev/raw/raw100	temp_0_2	3000
/dev/sdg13	/dev/raw/raw107	cust_0_2	5000
/dev/sdh7	/dev/raw/raw73	stok_0_3	7500
/dev/sdh8	/dev/raw/raw80	icom_0_3	3500
/dev/sdh9	/dev/raw/raw87	dcom_0_3	2500
/dev/sdh10	/dev/raw/raw94	ordl_0_3	8032
/dev/sdh11	/dev/raw/raw101	temp_0_3	3000
/dev/sdh12	/dev/raw/raw108	cust_0_3	5000
/dev/sdh13	/dev/raw/raw115	ctl2	1000
/dev/sdi6	/dev/raw/raw74	stok_0_4	7500
/dev/sdi7	/dev/raw/raw81	icom_0_4	3500
/dev/sdi8	/dev/raw/raw88	dcom_0_4	2500
/dev/sdi9	/dev/raw/raw95	ordl_0_4	8032
/dev/sdi10	/dev/raw/raw102	temp_0_4	3000
/dev/sdi11	/dev/raw/raw109	cust_0_4	5000
/dev/sdj6	/dev/raw/raw75	stok_0_5	7500
/dev/sdj7	/dev/raw/raw82	icom_0_5	3500
/dev/sdj8	/dev/raw/raw89	dcom_0_5	2500
/dev/sdj9	/dev/raw/raw96	ordl_0_5	8032
/dev/sdj10	/dev/raw/raw103	temp_0_5	3000
/dev/sdj11	/dev/raw/raw110	cust_0_5	5000
/dev/sdj12	/dev/raw/raw116	roll01	1500
/dev/sdj13	/dev/raw/raw117	sys	4000
/dev/sdk6	/dev/raw/raw76	stok_0_6	7500
/dev/sdk7	/dev/raw/raw83	icom_0_6	3500
/dev/sdk8	/dev/raw/raw90	dcom_0_6	2500
/dev/sdk9	/dev/raw/raw97	ordl_0_6	8032
/dev/sdk10	/dev/raw/raw104	temp_0_6	3000
/dev/sdk11	/dev/raw/raw111	cust_0_6	5000
/dev/sdl6	/dev/raw/raw112	log1	8032
/dev/sdl7	/dev/raw/raw113	log1	8032

APPENDIX B – INIT.ORA PARAMETER FILE

```
compatible = 9.0.1.3.0
control_files = (/dwork1/ctl1,/dwork3/ctl2)
#control_files = ($diskloc/ctl1,$diskloc/ctl2)
db_name = fstest
db_block_size = 2048
db_files = 100
db_block_buffers = 750000 # SGA 1.7 Gb
sort_area_size = 10485760
shared_pool_size = 25000000
shared_pool_size = 15000000
log_buffer = 1048576
parallel_max_servers = 50
recovery_parallelism = 40
dml_locks = 500
processes = 619
sessions = 619
transactions = 600
cursor_space_for_time = TRUE
undo_management = auto
UNDO_SUPPRESS_ERRORS = true
undo_retention = 60
UNDO_TABLESPACE = undo_ts
max_rollback_segments = 520
db_writer_processes = 1
dbwr_io_slaves = 10
```



Linux Filesystem Performance Comparison for OLTP
January 2004
Authors: Rajendra Kulkarni, Peter Schay

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
www.oracle.com

Oracle is a registered trademark of Oracle Corporation. Various product and service names referenced herein may be trademarks of Oracle Corporation. All other product and service names mentioned may be trademarks of their respective owners.

Copyright © 2004 Oracle Corporation
All rights reserved.