White Paper

Oracle9i on IBM ESS F20 (Shark)

Oracle Redo Logs and ESS RAID5 Volumes:

To log or not to log?

V1.1

Summary:

This paper will present our internal benchmark results on which we based our decision to put our redo logs on IBM ESS (Shark) RAID5 volumes instead of JBOD (Just Bunch Of Disks). Yes, it sounds crazy and completely against old wisdom that is teaching us: "Do not put any write intensive files, especially redo logs on RAID5 volumes (or you'll be late for supper ;-)".
The bottom line of our 'benchmark' is that redo logs on ESS RAID5 volumes performed slightly better than JBOD volumes with Oracle built-in redo log mirroring.
We're not trying to proof anything with this paper; we think that hardware RAID1 is still the way to go with Oracle redo logs. Unfortunately, as you probably know the IBM ESS doesn't support RAID1. You can either configure rank of disks as volume(s) on RAID5 or JBOD. So, the question is where to put the redo logs? How about using OS LVM (Logical Volume Manager) to mirror disks in JBOD configuration as some people suggested on various forums? Well, we're running Oracle on Windows 2000 Advanced Server and OS mirroring of JBOD volumes was not an option for us (the reasons why we don't trust W2K fault tolerant driver is beyond the topic). We also think that using SW RAID volumes in general purpose operating systems with built-in RAID support, is not suitable for production databases and should be avoided. (By the way, we think this is true for any OS and any level of SW RAID, not just for MS stuff – saving money on HW vs. SW RAID is really bad investment!)
Finally we left with two options; to configure one rank with JBOD and use Oracle mirroring or to create redo logs on RAID5 volumes. Since we didn't find any trustworthy information from user community on the Net (beyond the rumors) we decided to do our own tests.
If you're new to terms such as SAN, Fibre Channel, or want to learn more about IBM ESS – Enterprise Storage Server, then you should point your browser to: http://storage.ibm.com/ess and http://redbooks.ibm.com.

Aleš Kavšek
OCP-DBA, MCSE

March, 2002

DISCLAIMER: (or what my layer would make me say...if I had one):
Information provided in this document is provided "AS IS" without warranty of any kind. The information contained in this paper represents the current view of author and as such is not in any connection with author current employer or anyone else.

You can copy or distribute this document freely in any form under the following conditions:

- all text must be copied without modification and all pages must be included
- this paper may not be distributed for profit

All trademarks and copyrights mentioned in this white paper are acknowledged -- they're the property of their respective owners.

And last but not least author assumes no responsibility for errors in this paper, omissions or for damage resulting from the use of the information contained herein. (Sounds familiar?:-)
All procedures and precaution has been taken in the preparation of this paper to avoid errors.

You can send comments to: kavsek@yahoo.com

0. Introduction

Hard disk subsystems are still the target number one when it comes to database performance tuning. Solid state disks (SSD) will eventually solve *some* of today I/O bottlenecks (and definitely bring us some new headaches), however right now they're way too expensive for even medium sized databases (~50 GB) not to mention multi terabyte storage systems. Nevertheless, only few technology sectors prospered so fast as storage technology did in past three to four years.
One such technology field is certainly SAN – Storage Area Network. The fact that storage solutions build around SAN are today state of the art components in *overall enterprise database architecture* also means that you're at the same time faced with new, *'bleeding edge'* technology propelled with hype from vendors and resellers.

There are some *old questions* that need *new answers*[1]. For example:

- What is the *optimal database layout* for particular vendor SAN solution?
- How to *partition storage* under SAN?
- Where to put the *most I/O intensive files*, such as *redo logs*, temp and rbs segments?

The last question was of particular interest to us. IBM ESS F20 supports logical volumes on RAID5 or JBOD (Just Bunch Of Disks). According to information we found in IBM Redbooks the IBM position on the issue is clear – you should avoid JBOD and use RAID5 volumes whenever is possible. Right from the beginning we had some reservations with this recommendation. Why?

Storage vendors tend to overestimate the performance of their products; this is especially true for RAID5 volumes. We all know that RAID5 volumes are good choice for predominantly read intensive environments and bad for write intensive one. Our production databases are DSS/DW oriented with both read/write intensive daily jobs.

It's best to illustrate the problem to those of you who are not familiar with RAID5 with simple example. Lets say that you wish to update one row in the database table that is stored on RAID5 volume (for the sake of simplicity we assume plain table; no indexes on that table, row chaining etc.). Oracle will read the block with that row in db cache, after updating the row in memory *lazy write process* (database writer) will eventually write the block back to disk. When RAID controller intercepts the call to write the block back to disk it calculates the parity information and writes *two blocks* of data to *two different* disks (data block + parity block). Read operations on RAID5 volumes doesn't have parity overhead, on contrary they can benefit from data striping (with one exception; in the case of disk failure which is part of the RAID5 volume, parity blocks must be read to recalculate the original data).

During the years storage vendors developed different techniques to minimize the overhead of RAID5, mostly with internal algorithms, queued writes and large caches (anything from several megabytes to several gigabytes). Despite of their efforts it's hard to believe that they can completely eliminate parity overhead during the write phase. It's a common sense then to be careful and not to believe everything you're told from storage vendors and resellers, no matter which brand they represent.

[1] At the time of writing this paper we could not find any usable information on the topic, neither from Metalink nor from TAR request.

One of the questions that we asked ourselves was: "Why did IBM abandon RAID1 in ESS design and implemented only RAID5 and JBOD?" (Obviously system architects who designed IBM ESS were very confident in their approach.)

On the other hand we found plenty of rumors about Oracle and ESS in newsgroups. I remember one post in the newsgroup comp.databases.oracle.server that was going like this:

Quote...
"...Don't let the IBM tech guys fool you – they will say "Why use RAID1 – our RAID5 is much faster with our "optimized" routine in ESS"...the truth is that when a disk crashes in a JBOD configured 8 pack they can not simply replace the disk in 5 min. This means extra work for the IBM tech guy – so they try to convince you to use RAID5 all the way..."
...end quote

Bizarre statements and rumors like this triggered decision to perform our own basic tests[2].

In the next section we'll present test environment and tools that we used.

[2] We must explicitly stress that our experience with IBM engineers working with us on setting up IBM ESS is quite the opposite. Not once did they try to persuade us to implement RAID5 instead of JBOD, on contrary they helped us to prepare test environment for us to experiment with both options.

1. Test environment and objectives

All tests were done with our scripts in *best effort fashion* and on our own, without Oracle or IBM interfering in any way! We don't pretend that our results are "*politically correct"* and we're sure that with more scientific approach (using statistical methods, sufficient number of test runs etc.) and more resources (time) results would be more accurate. However we believe that results of our tests served the main objective which was to find out if IBM ESS RAID5 volumes are suitable to handle workload of Oracle on-line redo logs versus JBOD.
It's completely up to you to accept our final conclusion or make your own - what works for us doesn't mean it'll work for you!!

For testing we used Dell Power Edge 6400, four way SMP server with two internal 18GB SCSI disks. We used Dell internal RAID controller to configure RAID1 with two internal disks. We made two partitions on 18GB logical volume; on C: partition we installed Windows 2000 AS (SP2), on D: partition we installed Oracle Software (RDBMS without database!). Then we installed two HBA (Host Bus Adapters) from QLogic, each adapter connected to separate fibre channel switch. Subsystem device driver (SDD) is needed if you're using more than one HBA in server to provide load balancing and failover, so we installed SDD as well. We used 4GT option on Windows 2000 AS to provide 3GB memory to user space.
Next we prepared and assigned some RAID5 and JBOD types of volumes on IBM ESS to Dell server. We needed those volumes for testing our scenarios. Last step in preparing test environment was installation of database on ESS volumes. No special tuning efforts has been done with Oracle RDBMS or Windows 2000 AS - more or less we kept all parameters values close to our experience so far with Oracle on Windows platform.

See table 1 for some technical details and figure 1 for an overview of our test environment.

Table 1.

Server:	model: Dell Power Edge 6400
	CPU : 4 x 900 MHz Xeon
	RAM : 4 GB
	HDD : 2 x internal 18GB SCSI disks (RAID1)
	NIC : 2 x Gigabit adapter (Intel)
	HBA : 2 x QLogic QLA2200 (Bios 1.61)
OS:	Windows 2000 Advanced Server SP2
	4GB Tuning switch in boot.ini
	SDD (Subsystem device driver)
Storage:	model : IBM ESS F20
	capacity:64 SSA disks (36 GB per disk)
	8 disks out of 64 are spare disks
FC Switch:	IBM SAN FC switch 2109 Model S16
	2 x 8 port switches
RDBMS:	Oracle9i EE Release 1 (9.0.1.2)
	Total size of SGA: 2.2 GB
	Database running in Archive Log mode
	Block size: 8 KB
	Redo log size: 500 MB

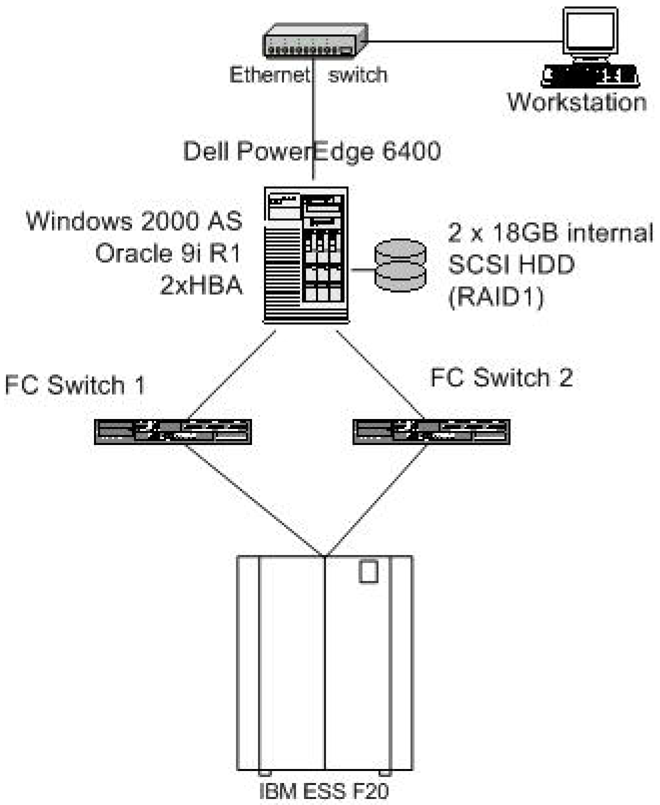


Figure 1.

Now let's see how we partitioned our eight ranks of disks in IBM ESS for this test. We configured 7 ranks as RAID5 and one rank as JBOD (J1, J2....) as you can see in figure 2. Only DATA1, DATA2, LOG1, LOG2, LOGARCH, J1, J2, J3 and J4 volumes were used in our tests.

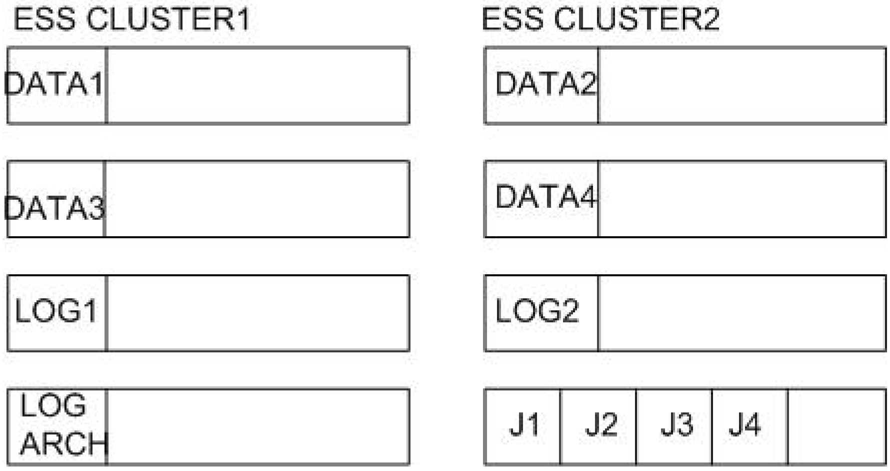


Figure 2.

In first scenario we placed redo logs on RAID5 volumes (LOG1, LOG2). We created four redo log groups with *one member* in each group. All four logs were of uniform size - 500MB. First group was on LOG1, second on LOG2, third on LOG1 and fourth on LOG2. Archive logs are stored on LOGARCH volume (RAID5).

In second scenario we placed redo logs on JBOD volumes J1, J2, J3 and J4. We again created four redo log groups but this time *with two members* in each group, letting Oracle mirror the redo logs. We put members from first group on J1 and J2, from second group on J3 and J4, the third group on J1 and J2 and fourth group on J3 and J4. Again all redo logs were of uniform size (500 MB). Archive logs are stored on LOGARCH volume (RAID5).

In both scenarios we placed database data files on DATA1 and DATA2 volumes, for archiving redo logs we choose LOGARCH volume, and for Undo tablespace volume DATA1.

As you can see we had to compromise a little bit due to some other considerations and planned usage for IBM ESS at our site. (Without doubt we could make some other layout of the volumes and Oracle database, but for the purpose of the test we decided to keep this one.)

So, in first scenario Oracle writes to only one redo log at the time on volume that is protected by RAID5 from a single disk failure. In second scenario we're using Oracle built-in mirroring support of redo logs as protection against single disk failure. That means that Oracle must write twice as much but on disks with no RAID5 overhead. One good thing about second scenario is better protection from software corruption of redo log files (two redo log files in a group offer better protection than only one as it's the case in first scenario).

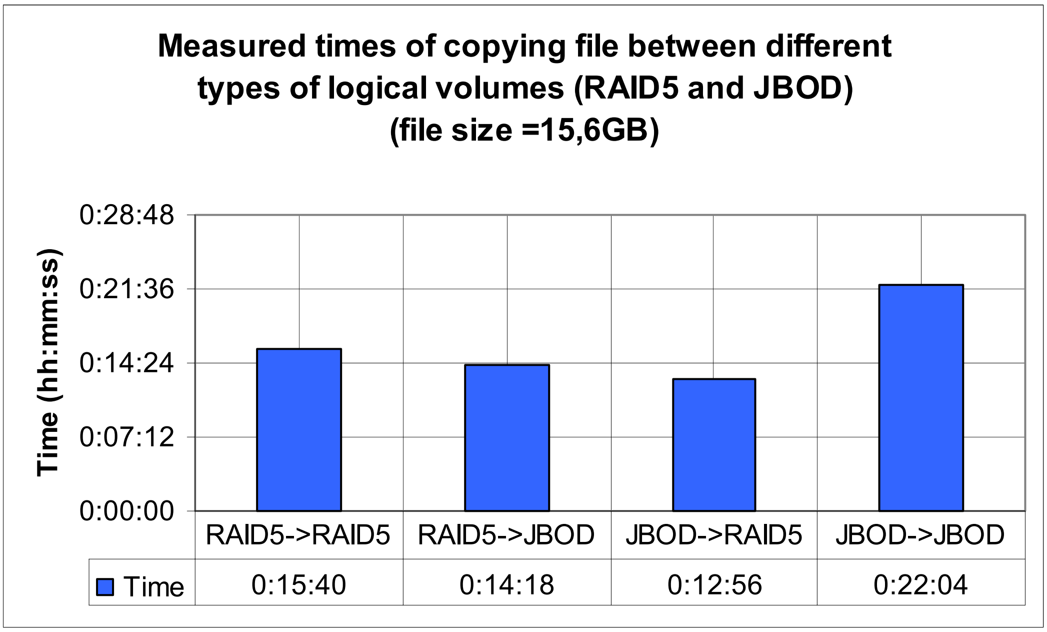
In the next section we'll present test results with short comments.

2. Test results

2.1. Test #1: Copying large file between different types of logical volumes

In first test we measured elapsed time transferring large (15,6GB) file between different kinds of volumes.

RAID5-> RAID5 ... transfer from DATA1 to DATA2
RAID5-> JBOD ... transfer from DATA1 to J1
JBOD-> RAID5 ... transfer from J1 to DATA1
JBOD-> JBOD ... transfer from J1 to J2



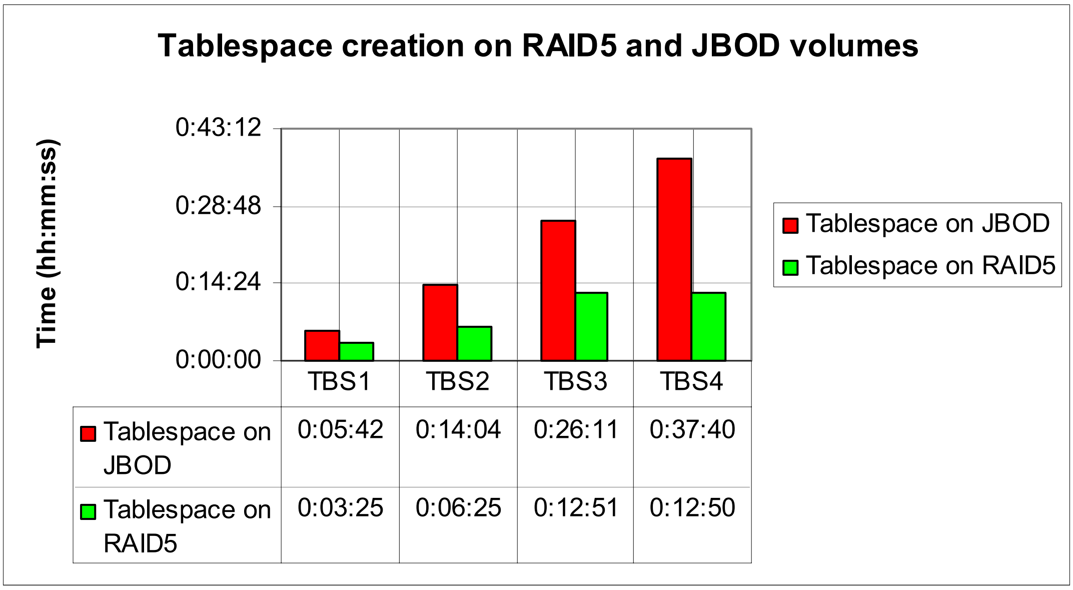
It seems that RAID5 doesn't have negative impact on serial writes compared to JBOD.

2.2. Test #2: Tablespace creation on RAID5 and JBOD volumes

In our second test we measured elapsed time creating tablespaces of different sizes on DATA2 volume (RAID5) and J2 volume (JBOD):

- TBS1 = 8 GB
- TBS2 = 16 GB
- TBS3 = 32 GB
- TBS4 = 2 x 16 GB concurrently

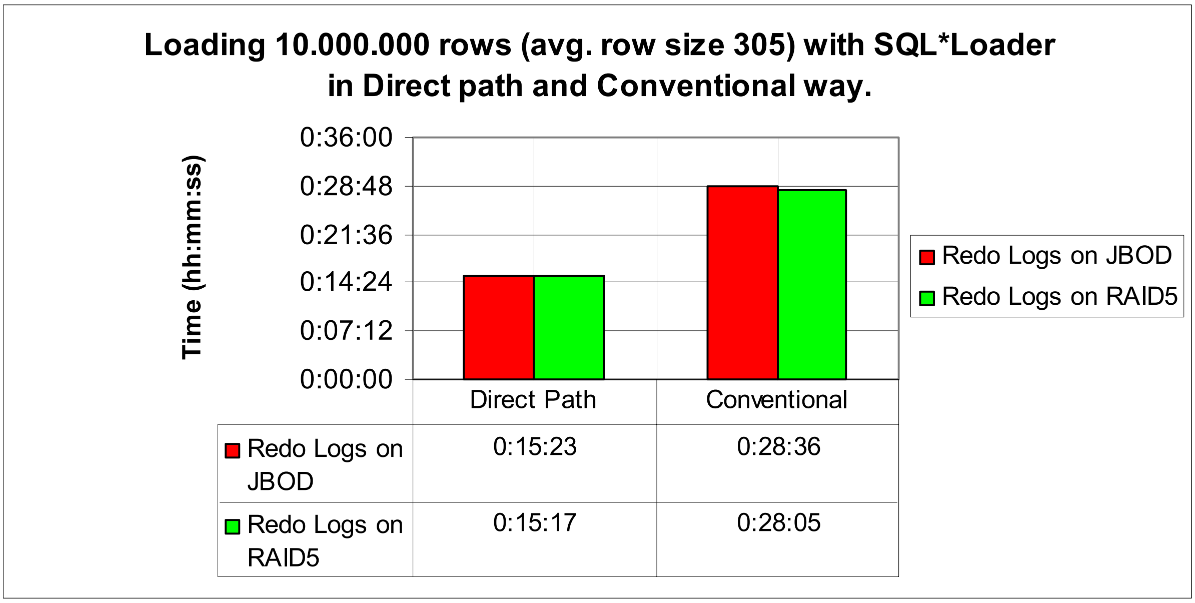
Clear winner in this test is RAID5. Serial write operations on ESS are obviously much faster on volumes configured under RAID5 than on plain disk. It took Oracle half the time to create tablespace on RAID5 volume than on single JBOD disk. We also performed test TBS4, creating two 16GB tablespaces concurrently at the same time, first on DATA2 volume and then on J2 trying to skew a little bit perfect sequent ional write nature during tablespace creation. Notice that duration time creating two 16GB tablespaces at the same time on RAID5 volume is almost the same compared to TBS3 test with single 32 GB tablespace (difference of a second). On the other hand it took considerably longer to create two 16GB tablespaces at the same time on J2 disk.



	TBS1	TBS2	TBS3	TBS4
🟥 Tablespace on JBOD	0:05:42	0:14:04	0:26:11	0:37:40
🟩 Tablespace on RAID5	0:03:25	0:06:25	0:12:51	0:12:50

2.3. Test #3: Loading data with SQL*Loader (Direct path and Conventional way)

We loaded 10 mio. records in a table with 88 columns. Average row size in table was 305 bytes. Data was loaded from a file (10 mio. rows 498 bytes in length) located on a local SCSI disk. During direct path load we used parameter UNRECOVERABLE (to bypass redo logs) and ROWS=100000 in control file. For conventional load we used parameter ROWS=100000 and loaded data in RECOVERABLE mode.



	Direct Path	Conventional
🟥 Redo Logs on JBOD	0:15:23	0:28:36
🟩 Redo Logs on RAID5	0:15:17	0:28:05

2.4. Test #4: Update / Insert / Delete / Drop Column on table with 10 mio. rows

Update1 Updating table with 10 mio. rows (avg. row size=305 bytes) by issuing 12 consecutive update statements. Table was stored on DATA2 volume and it contained some data for a year 2001. We committed after updating data for each month (approx. every 800000 records ~ 240 MB).

```
DECLARE
 i number;
BEGIN
 For i IN 1..12 LOOP
  UPDATE ess_test_table1 SET yearmonth=year||month
         WHERE year=2001 and month=i;
  commit;
 END LOOP;
END;
/
```

Update2 Updating two exact tables with 10 mio. records (avg. row size=305 bytes). One table was stored on DATA1 volumes (ESS Custer 1) and second one on DATA2 volume (ESS Cluster 2). We basically used two scripts like the one we used for Update1 test. Undo tablespace is located on DATA1, this volume is obviously hotspot in this test.

Insert1 During this test we measured elapsed time copying data from one table (DATA2) to another (DATA1) with script below:

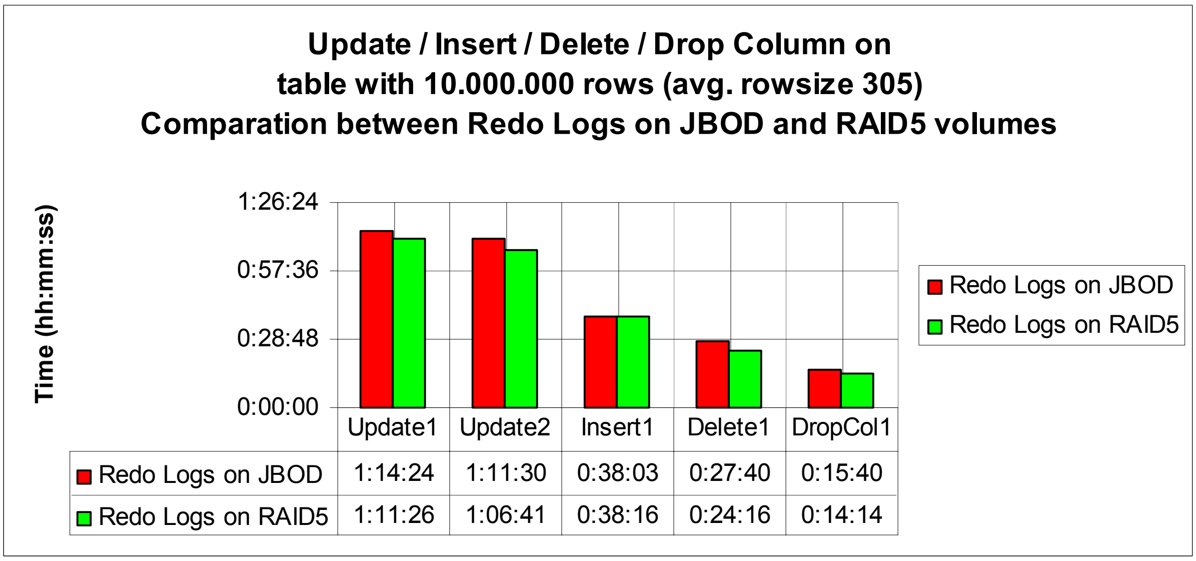
```
DECLARE
 i number;
BEGIN
 FOR i IN 1..12 LOOP
    INSERT INTO ess_test_table2 select * from
                ess_test_table1
    WHERE year=2001 and month=i;
    commit;
 END LOOP;
END;
/
```

Delete1 We deleted all rows from table with 10 mio. records, committing after every 200000 records.

DropCol1 In this test we dropped one column from a table with 10 mio. records.

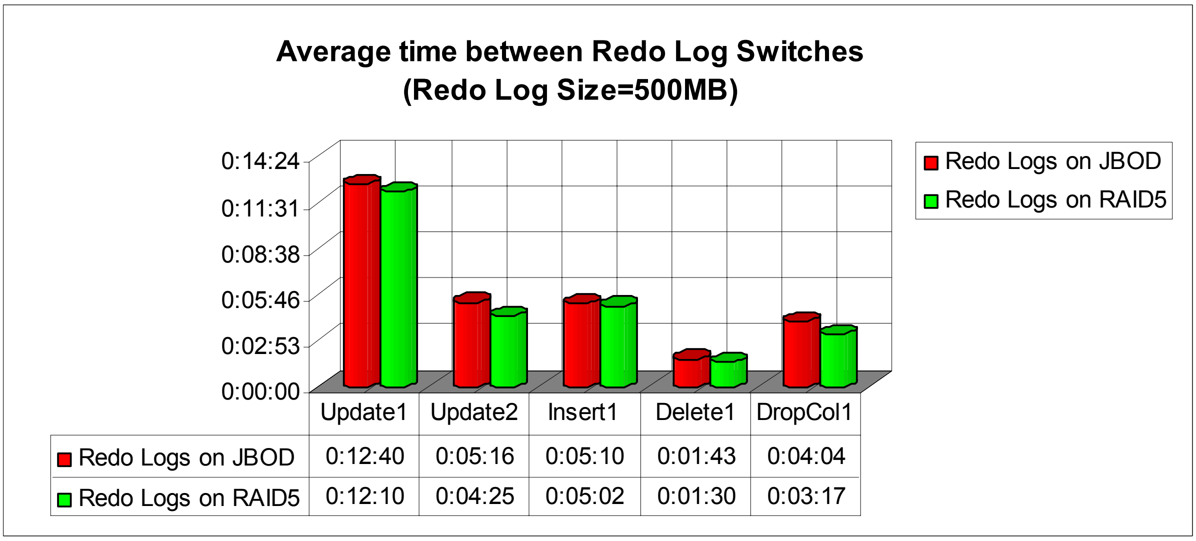
```
ALTER TABLE ess_test_table1
  DROP COLUMN areacode CHECKPOINT 200000;
```

The results of the tests are presented in the following chart;



	Update1	Update2	Insert1	Delete1	DropCol1
🟥 Redo Logs on JBOD	1:14:24	1:11:30	0:38:03	0:27:40	0:15:40
🟩 Redo Logs on RAID5	1:11:26	1:06:41	0:38:16	0:24:16	0:14:14

During this set of tests we captured data from V$LOG_HISTORY. On the next chart you can see average time between Redo Log switches for particular test (as you can remember the size of the redo log is 500 MB).



	Update1	Update2	Insert1	Delete1	DropCol1
🟥 Redo Logs on JBOD	0:12:40	0:05:16	0:05:10	0:01:43	0:04:04
🟩 Redo Logs on RAID5	0:12:10	0:04:25	0:05:02	0:01:30	0:03:17

As you can see for yourself there is not much difference between the two configurations. Oracle instance with redo logs on RAID5 performed slightly better than instance using Oracle built-in mirroring of redo logs on JBOD disks. (Remember that we used only *one redo log per group* in RAID5 configuration and *two redo logs per group* on JBOD configuration -- it means that Oracle must write twice the amount of data on database instance with redo logs on JBOD compared to hardware RAID5). If you can cope with the fact that *one redo log* in a group doesn't offer the same level of protection as two (or more) group members, then you don't need to hesitate with putting redo logs on ESS RAID5 volume. They'll perform well. Since we didn't experiment with mirroring with OS LVM instead of Oracle, we can only speculate that the results would not be considerably different. We're sure however that with RAID1 implemented in ESS instead of using Oracle built-in mirroring the results would probably be in favor of putting write intensive files on ESS mirrored volumes.

2.5. Some data collected with Performance Monitor during the tests

These results must be taken with caution! They're only relevant to our setup environment with all constraints (type of server, operating system, etc.) and as such can not indicate maximum ESS throughput. You should contact IBM and request official benchmarks for your target platform! We included data collected with performance monitor in this paper for your convenience and your own interpretation!

Copying 16GB file from RAID5 volume to JBOD volume

	JBOD			RAID5 (6+N)		
	MIN	AVG	MAX	MIN	AVG	MAX
Current Disk Queue Length	0	1	2	0	1	1
Disk Read Bytes / s	0	0	0	12228837	18873154	24838210
Disk Reads / s	0	0	0	186,6	288,1	379
Disk Write Bytes / s	12250360	18869968	22555036	0	0	0
Disk Writes / s	188,8	290	346,6	0	0	0

Copying 16GB file from JBOD volume to RAID5 volume

	JBOD			RAID5 (6+N)		
	MIN	AVG	MAX	MIN	AVG	MAX
Current Disk Queue Length	0	1	1	0	1	2
Disk Read Bytes / s	18192832	21463886	23907572	0	0	0
Disk Reads / s	277,6	327,5	364,8	0	0	0
Disk Write Bytes / s	0	0	0	16709757	21600546	22614014
Disk Writes / s	0	0	0	255,5	330,37	346

Copying 16GB file from JBOD volume to JBOD volume on the same ESS cluster

	JBOD			JBOD		
	MIN	AVG	MAX	MIN	AVG	MAX
Current Disk Queue Length	0	1	1	0	1	2
Disk Read Bytes / s	7680859	12689878	17851986	0	0	0
Disk Reads / s	117	193,6	272,4	0	0	0
Disk Write Bytes / s	0	0	0	7750489	12695820	17657888
Disk Writes / s	0	0	0	119,2	194,3	270

Tablespace creation on JBOD volume (8GB, 16Gb and 32 GB tablespace)

	JBOD			RAID5 (6+N)		
	MIN	AVG	MAX	MIN	AVG	MAX
Current Disk Queue Length	0	1	2			
Disk Read Bytes / s	0	0	0			
Disk Reads / s	0	0	0			
Disk Write Bytes / s	13213649	20008495	24963542			
Disk Writes / s	13	20,7	27,4			

Deletion of 10 mio. records from table on volume DATA1 (see test Delete1 on page 10).

	I/O on DATA1 (RAID5)*			
	LOGS ON JBOD		LOGS ON RAID5	
	AVG	MAX	AVG	MAX
Current Disk Queue Length	38	254	46	254
Disk Read Bytes / s	2062333	3648671	2077978	4092670
Disk Reads / s	238	363,6	238,4	422
Disk Write Bytes / s	7318082	19542863	7813561	20106299
Disk Writes / s	883,6	2380,4	943	2448,7

	I/O on volumes with REDO LOGS			
	AVG	MAX	AVG	MAX
Current Disk Queue Length	0,5	3	0	2
Disk Read Bytes / s	10878449,3	18175258,3	14993374	19806520
Disk Reads / s	10,4	17,3	14,4	18,7
Disk Write Bytes / s	5126193,2	6953422,4	6058939	8752824
Disk Writes / s	63,76	92	82,2	115,9

**On volume DATA1 is also undo tablespace so we can expect pretty much random I/O on that volume.*