



An Oracle White Paper
December 2013

VARCHAR and SecureFiles CLOB: A Comparative Analysis

Introduction	1
Summary.....	2
Test Description	3
Single Byte – US7ASCII	4
Single User Insert Results	4
Single User Select Results	6
Concurrent Users Insert Results.....	7
Concurrent Users Select Results.....	8
Multi-Byte – UTF8	9
Single User Insert Results	9
Single User Select Results	11
Concurrent Users Insert Results.....	12
Concurrent Users Select Results.....	13
Multi-Byte – UTF8 + ARCHIVELOG	14
Single User Insert Results	14
Single User Select Results	14
Concurrent Users Insert Results.....	15
Concurrent Users Select Results.....	15
Multi-Byte – UTF-8 + ARCHIVELOG + FLASHBACK	16
Single User Insert Results	16
Single User Select Results	16
Concurrent Users Insert Results.....	17
Concurrent Users Select Results.....	17
Hardware and Software Details	18
Conclusion	19
Appendix	20
Configuration: init.ora values	20
Graph Data.....	21

Introduction

The SecureFiles¹ feature was introduced in Oracle Database 11g as a major rewrite of LOB code to address performance and scalability problems associated with BasicFiles LOB in prior releases. Customers are widely using SecureFiles CLOB in their applications and are happy with the performance, but at times customers complain that SecureFiles CLOB performance is much slower than VARCHAR. Most often this wide performance gap is seen as one of the following five incorrect comparison cases:

- VARCHAR (Multi-byte) compared to SecureFiles IN-ROW CACHE (Multi-byte)
- VARCHAR compared to SecureFiles IN-ROW NOCACHE
- VARCHAR compared to SecureFiles OUT-ROW CACHE
- VARCHAR compared to SecureFiles OUT-ROW NOCACHE
- VARCHAR + FLASHBACK compared to SecureFiles + FLASHBACK

This paper shows how close SecureFiles CLOB performance is to VARCHAR's, illustrating with five result sets under four different test environments, listed in Table 1 (below).

¹ The terms SecureFiles LOB, SecureFiles, and CLOB are used interchangeably throughout this document and refer to a column defined as CLOB of type SECUREFILE, unless otherwise specified.

Summary

The experiments detailed in this paper lead to the following observations:

SecureFiles IN-ROW CACHE configuration is a fair comparison to VARCHAR, and it is slower by only a 5% range for data sizes less than 3500 bytes.

SecureFiles LOB data is moved out of table segment around 3500 bytes² and stored in LOB segment. Accessing LOB segment data is by default (nocache) using physical I/O and will thus be slower compared to VARCHAR (which by default is cached).

SecureFiles INSERT with Flashback option turned ON can show a big performance gap compared to VARCHAR performance depending on the settings of SecureFiles IN or OUT ROW and CACHE or NOCACHE storage option. This is because the Flashback option requires additional I/O on table or LOB segment blocks. This additional I/O usually manifests as wait events in the Automatic Workload Repository, namely “cell single block physical read,” “flashback log file sync,” “direct path read” and “direct path write.”

Table 1 below summarizes the test environments and result sets.

TABLE 1. TEST ENVIRONMENTS

TEST ENVIRONMENT	RESULT SETS
Single byte (US7ASCII)	VARCHAR
Multi-byte (UTF8)	SF-CLOB-IN-ROW-CACHE (SecureFiles-in-cache) SF-CLOB-IN-ROW- NOCACHE (SecureFiles-in-nocache)
Multi-byte (UTF8) + ARCHIVELOG ON	SF-CLOB- OUT-ROW -CACHE (SecureFiles-out-cache) SF-CLOB- OUT-ROW - NOCACHE (SecureFiles-out-nocache)
Multi-byte + ARCHIVELOG ON + FLASHBACK ON	

The remainder of the paper first explains the performance test, followed by detailed results arranged into one section per each environment with subsections for result sets. This is followed by details on the hardware and software setup and the raw data from each experiment in the Appendix.

² This LOB data movement happens around 1750 characters when a using a multi-byte (UTF8) character set, in which storing 1750 characters takes 3500 bytes.

Test Description

This performance test consists of a simple table with two columns, the first column is defined to hold document id (primary key) and the second column is defined to hold document content in VARCHAR or SF CLOB with four different LOB storage options as mentioned in Table 1.

Example:

```
create table DOC_VAR_TAB (doc_id number, doc_content VARCHAR)
create unique index doc_tab_idx on doc_var_tab(doc_id);

or

create table DOC_CLOB_TAB (doc_id number, doc_content CLOB) LOB
(doc_content) store as &lobtype_opt (&inout_opt &cache_opt);
create unique index doc_tab_idx on doc_clob_tab(doc_id);
```

Each test consists of single and multiple users doing inserts and selects as follows:

1. Single user insert test – one user doing insert of one million rows (commit after each row).
2. Single user select test – after the above insert test is done, one user doing one million times single row random selects (out of the above one million inserted rows) using the primary key column (doc_id).
3. Concurrent (8 user) insert test – eight users doing concurrent insert of one million rows into the same table (i.e. each user doing 25,000 row inserts).
4. Concurrent (8 user) select test – eight users doing concurrent single row random selects.

These tests are repeated for data sizes 70, 1750, 3500 and 4000 bytes and the results are summarized in a series of stacked bar chart graphs.

For each test, the first chart provides the DB time³ and IO time⁴ from the Automatic Workload Repository (AWR) report. Each vertical bar represents the DB time minus the direct path I/O, DBTime-io (**labeled green**) stacked with the direct path I/O time when appropriate **labeled blue** dpw_iowait (for inserts) or dpr_ioread (for selects).

The second chart provides total CPU resource usage expressed as a percentage. Actual elapsed time, DB time, IO wait times are provided in the appendix section for reference.

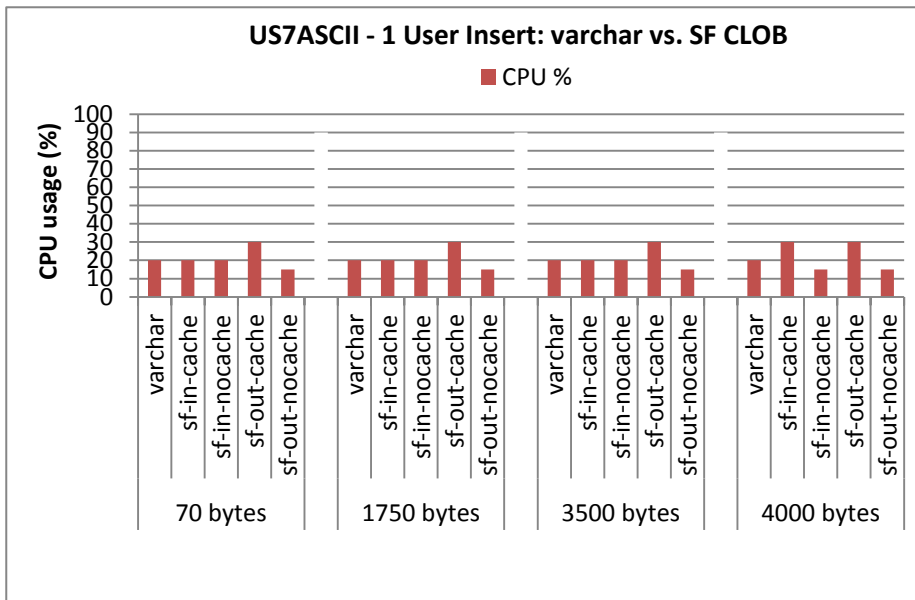
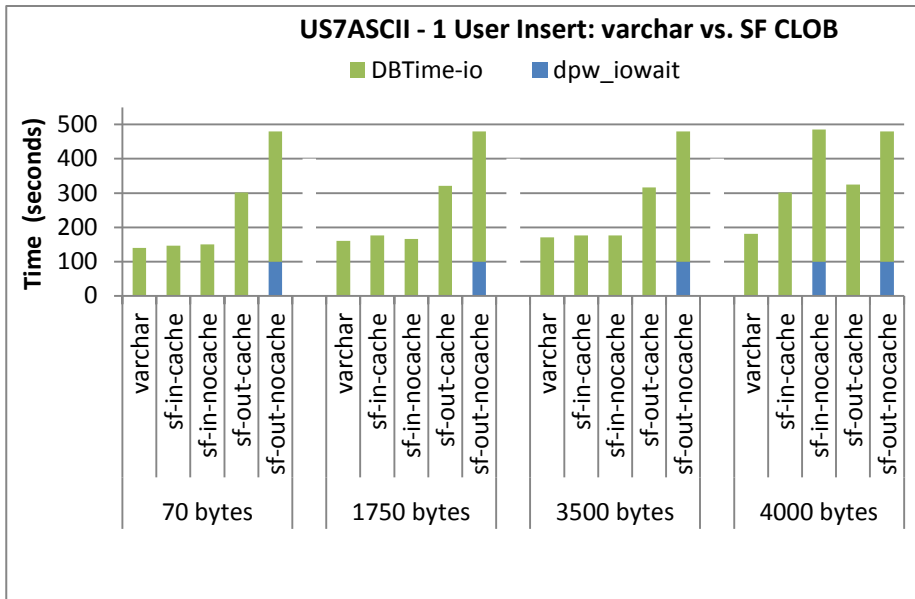
³ DB time = total time spent executing user statements, i.e., CPU time, I/O time, and non-idle wait time.

⁴ IO time = as reported under wait events such as “cell single block physical read,” “flashback log file sync,” “direct path read” and “direct path write.”

Single Byte – US7ASCII

Single User Insert Results

VARCHAR and SF-IN-CACHE CLOB performance and resource usage is almost the same for data sizes of up to 3500 bytes. For other data sizes and configurations the slowdown is caused mainly by LOB segment space allocation and/or direct path write waits.



Notes: Caching and Sizes

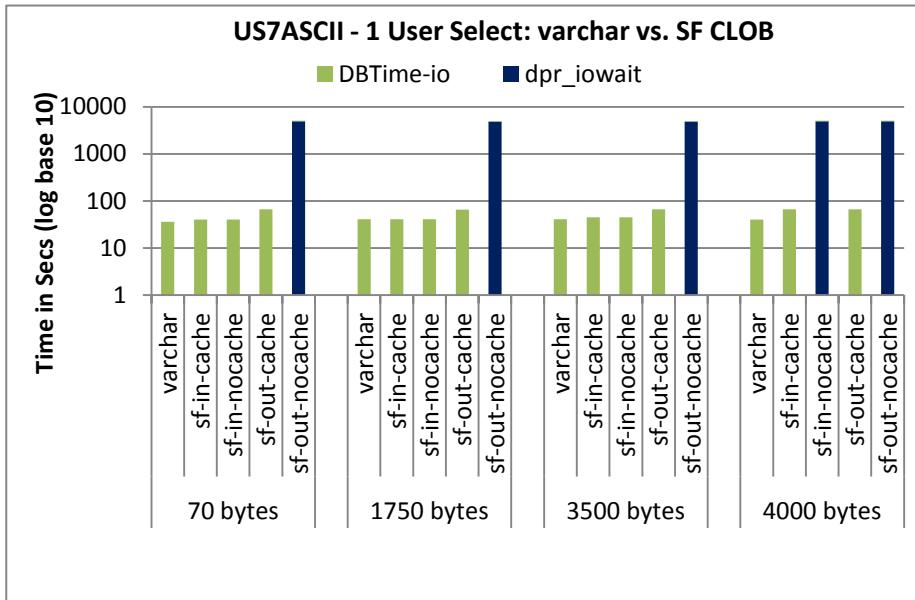
With sf-in-*, if LOB data is less than 3500 bytes, it is always cached, because data is stored in table row segment, irrespective of CACHE or NOCACHE option.

TABLE 2. TABLE OR LOB SEGMENT SIZE IN MB

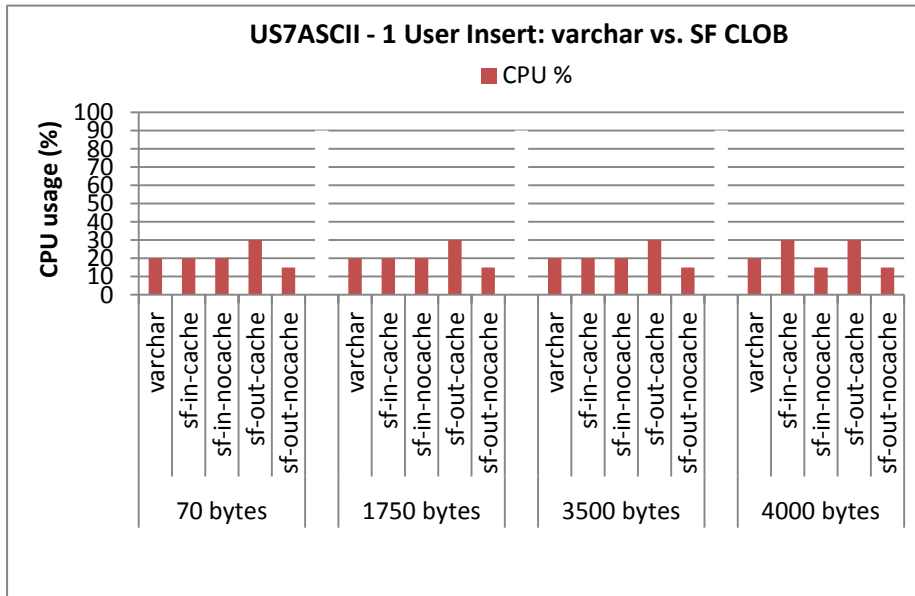
TYPE	70	1750	3500	4000
varchar	100	2000	4000	7900
sf-in-cache	130	2000	4000	8260
sf-in-nocache	130	2000	4000	8260
sf-out-cache	8260	8260	8260	8260
sf-out-nocache	8260	8260	8260	8260

With sf-out-*, LOB data is always stored in LOB segment and db_buffer cache is used for buffering only if LOB storage option CACHE is specified. See Table 2.

Single User Select Results

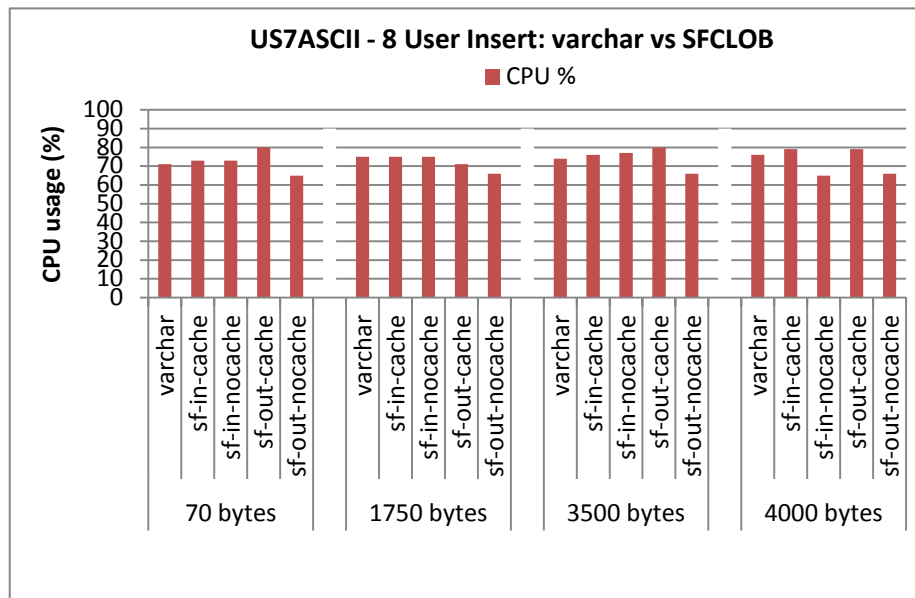
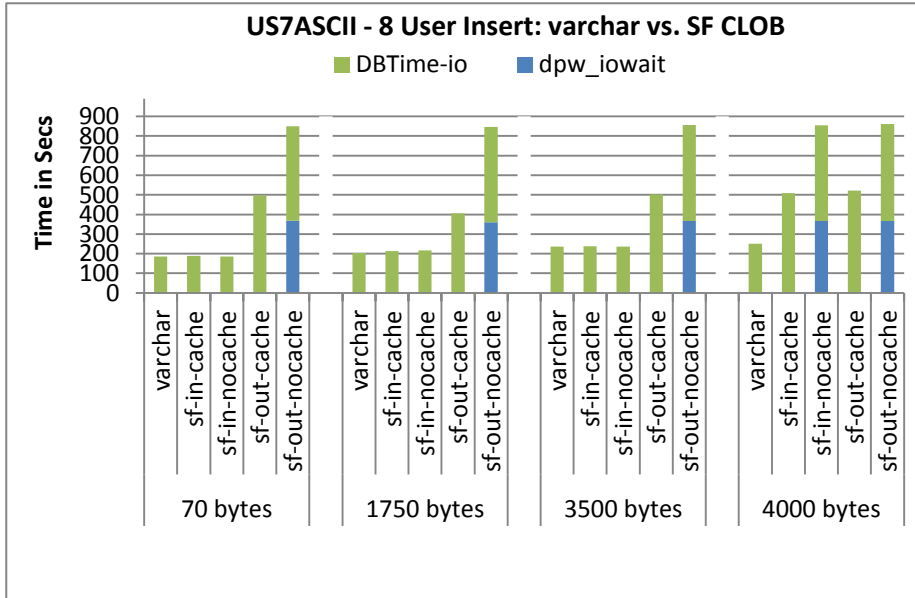


CLOB select performance is comparable to VARCHAR, except for the NOCACHE option, for which a major portion of the time is spent in doing “direct path read.”



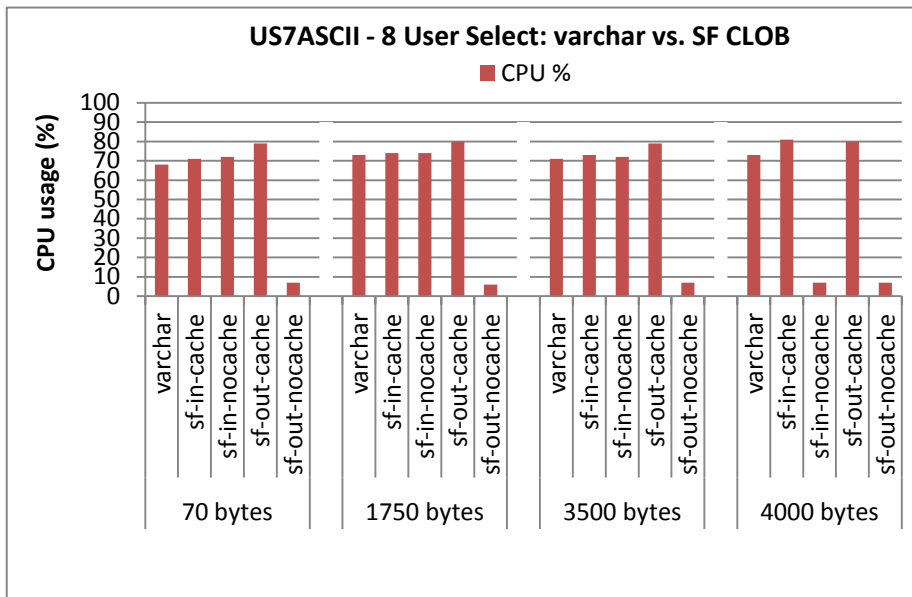
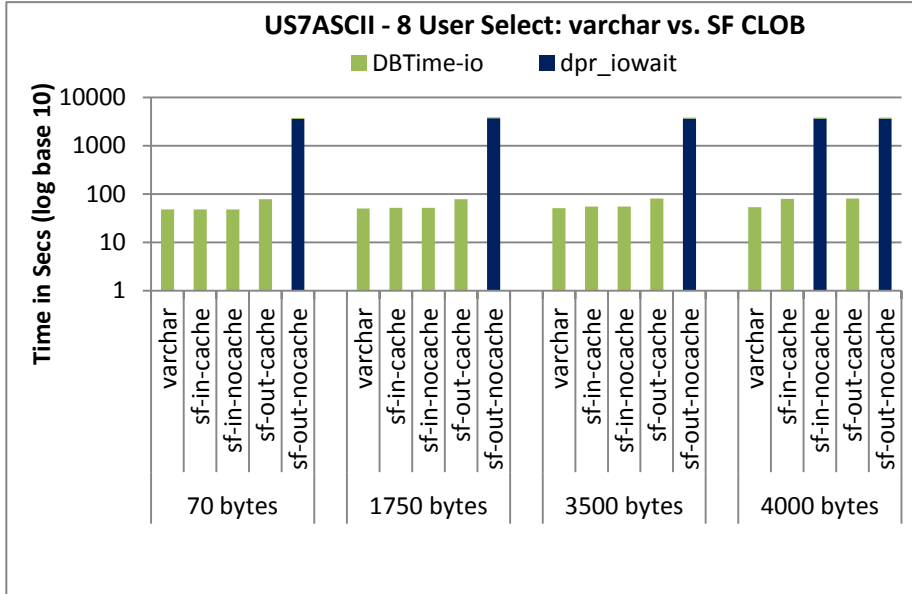
Concurrent Users Insert Results

Results for 8 users are similar to single user results. CPU usage is higher because of higher concurrency.



Concurrent Users Select Results

Results for the 8 user select test are similar to single user results.



Multi-Byte – UTF8

An interesting data point in this test is that LOB data storage takes almost two times the space compared to US7ASCII (single-byte) test (See Table 3). This makes even sf-in-nocache 3500 bytes configuration to be stored OUT-ROW.

Single User Insert Results

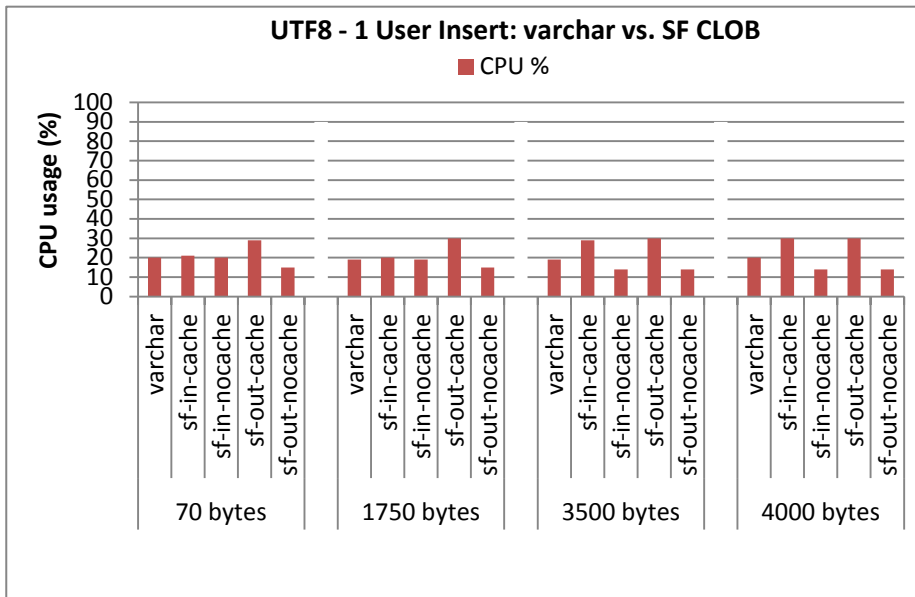
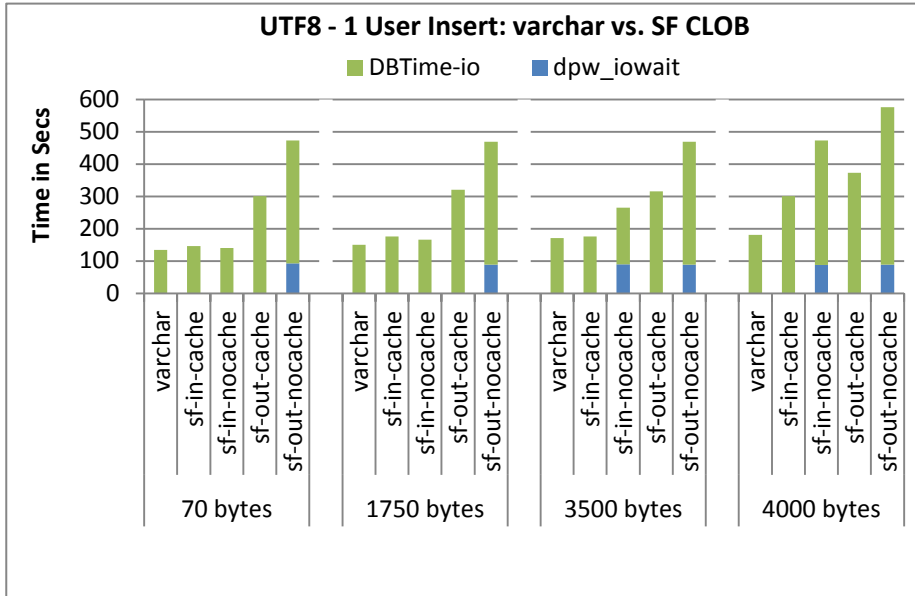
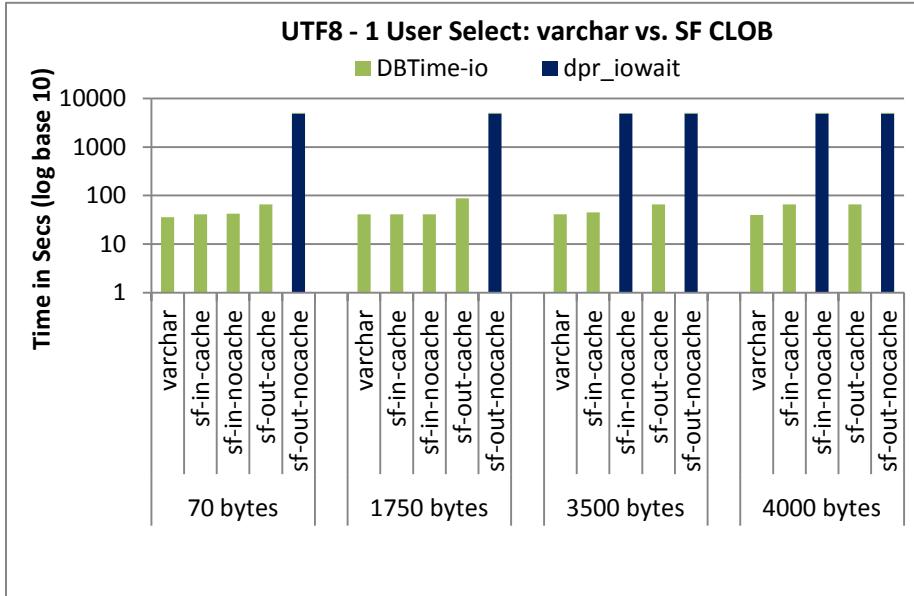


TABLE 3. TABLE OR LOB SEGMENT SIZE IN MB

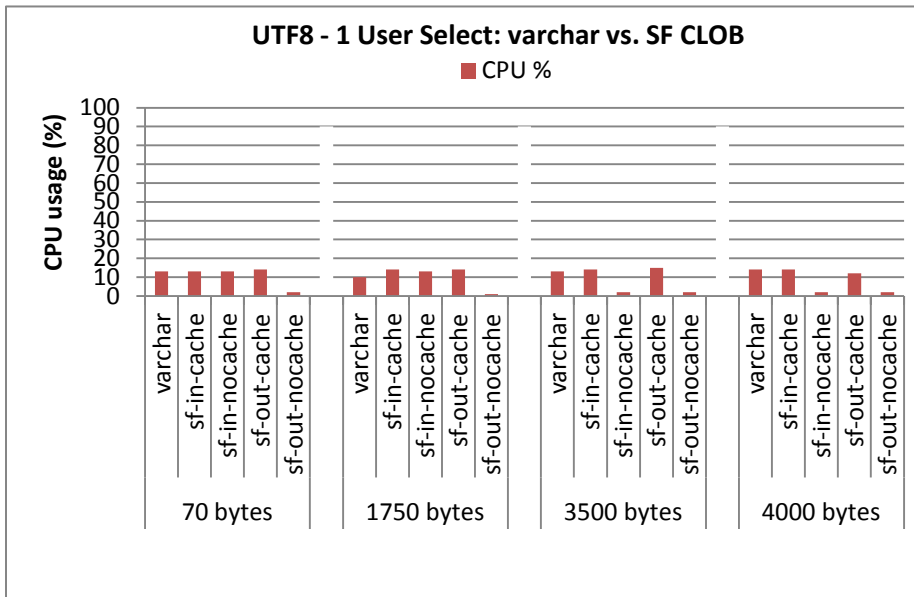
TYPE	70	1750	3500	4000
varchar	100	2000	4000	7900
sf-in-cache	200	4000	8260	8260
sf-in-nocache	200	4000	8260	8260
sf-out-cache	8260	8260	8260	8260
sf-out-nocache	8260	8260	8260	8260

Single User Select Results

CLOB select performance is comparable to VARCHAR, except for the NOCACHE option for which a major portion of the time is spent in doing “direct path read.”

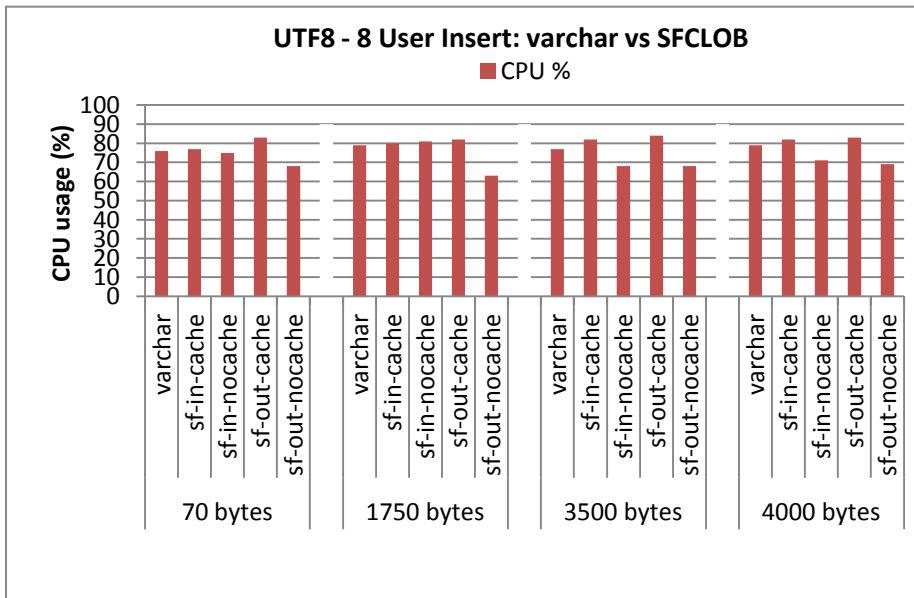
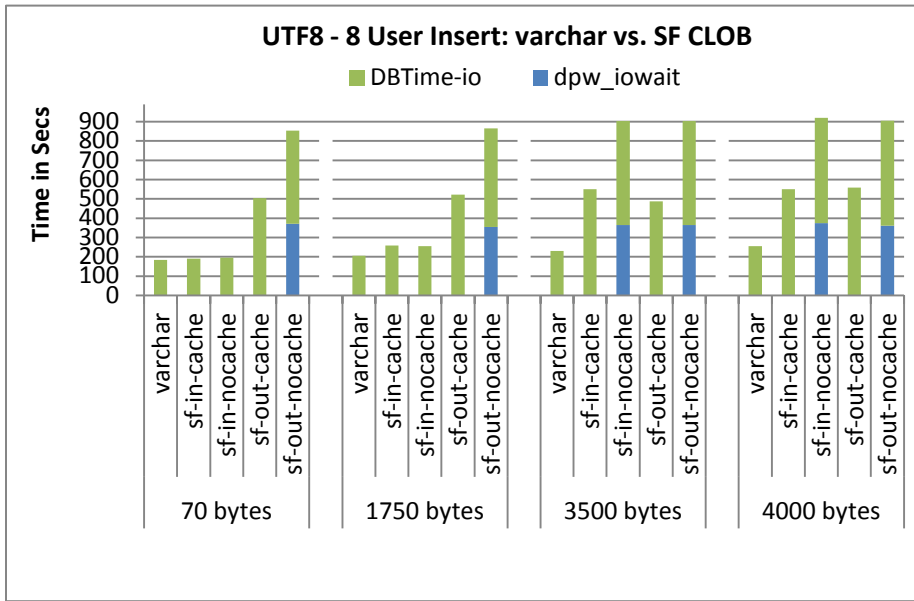


For NOCACHE, CPU usage is lot less compared to other configurations, because of I/O waits.

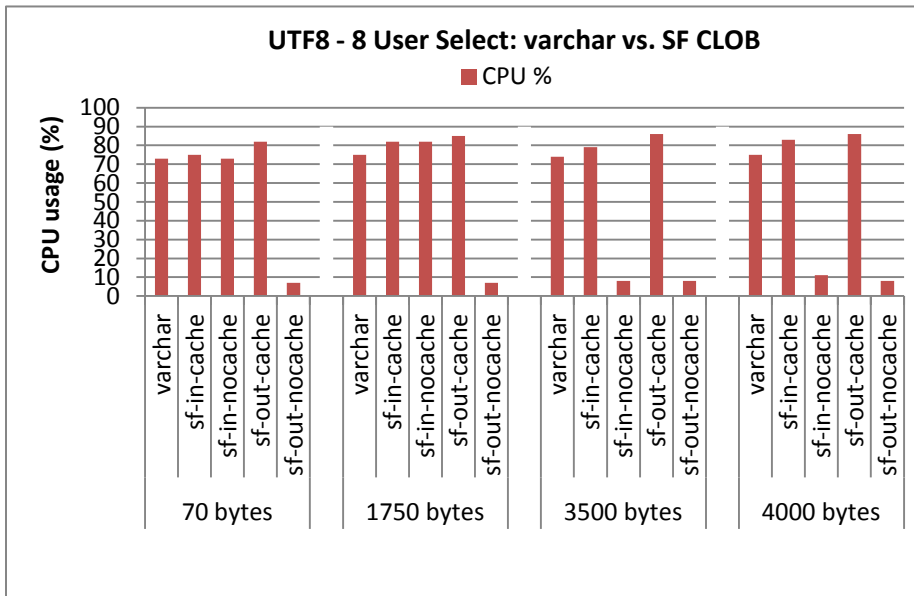
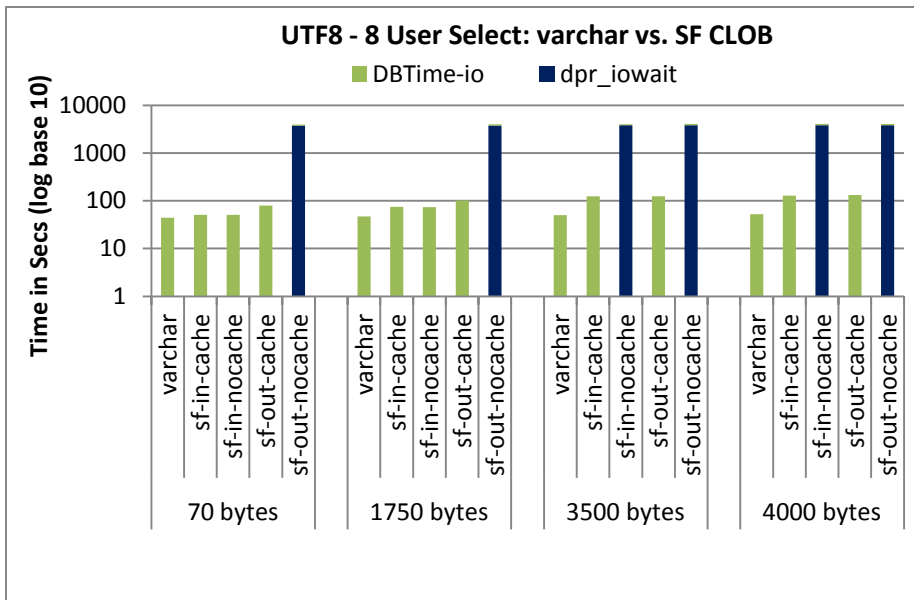


Concurrent Users Insert Results

Results for 8 users are similar to single user results. CPU usage is higher because of higher concurrency.

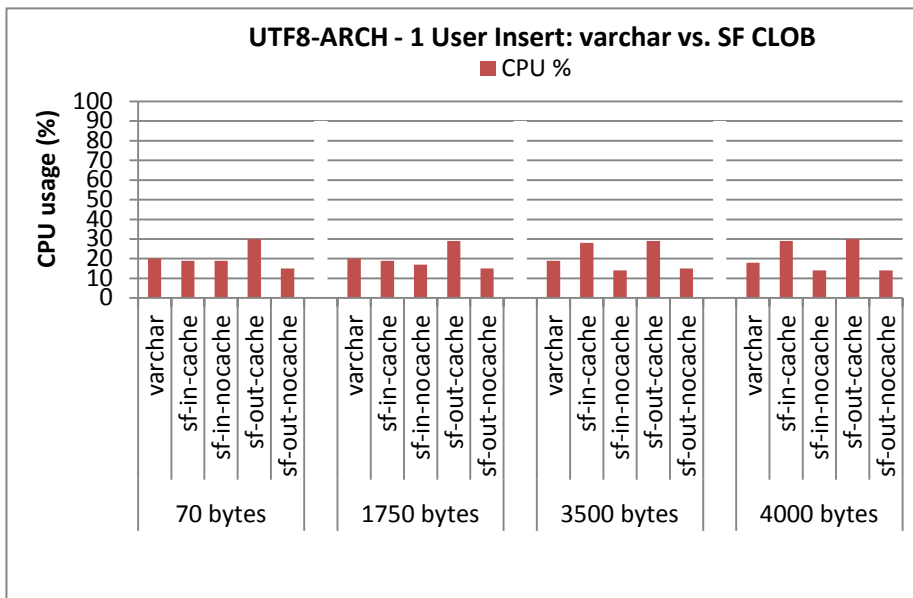
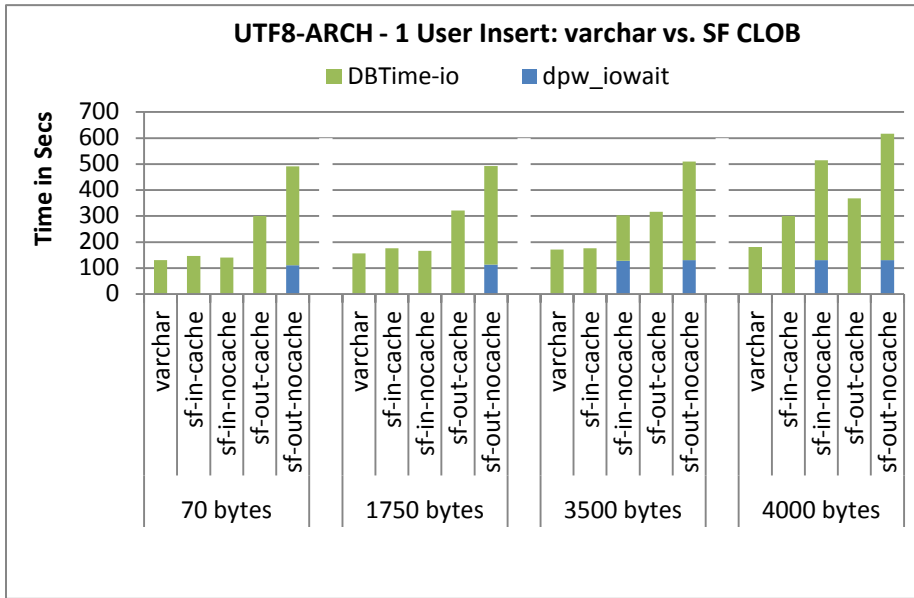


Concurrent Users Select Results



Multi-Byte – UTF8 + ARCHIVELOG

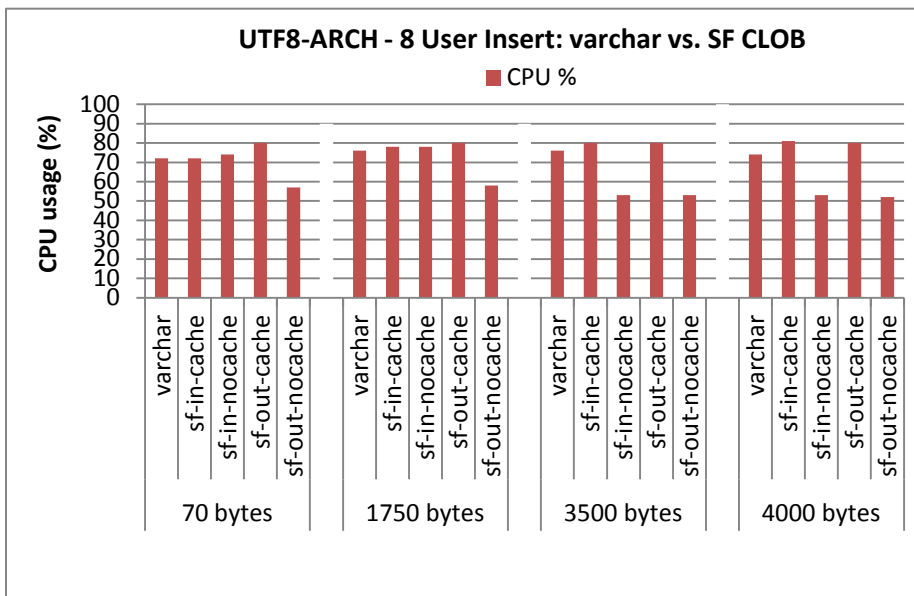
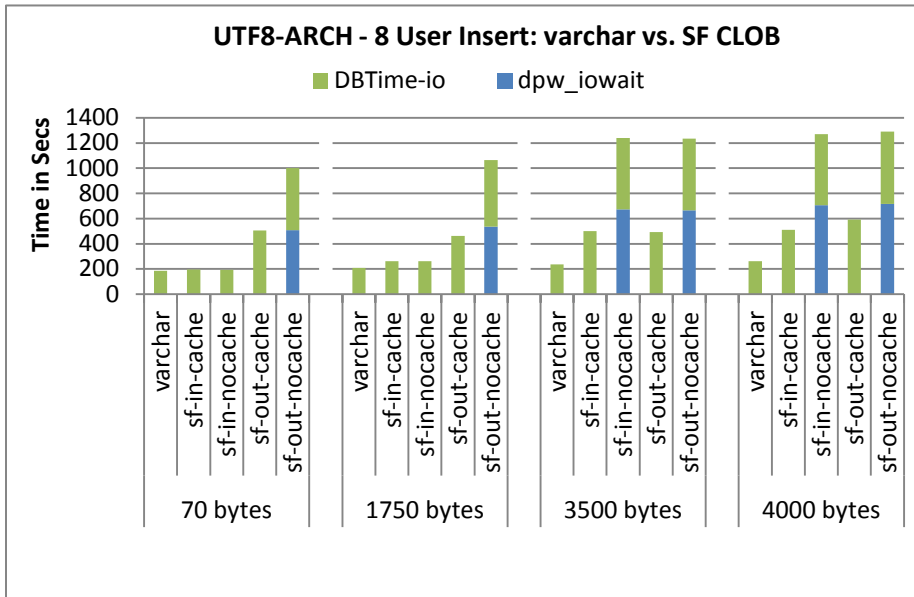
Single User Insert Results



Single User Select Results

Charts omitted because the results are very similar to UTF8 no archivelog single user selects.

Concurrent Users Insert Results



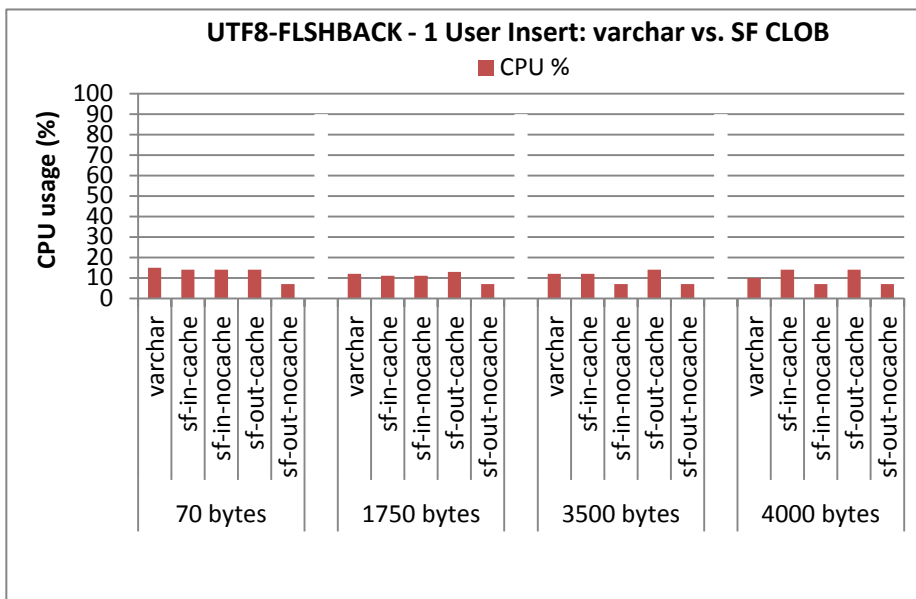
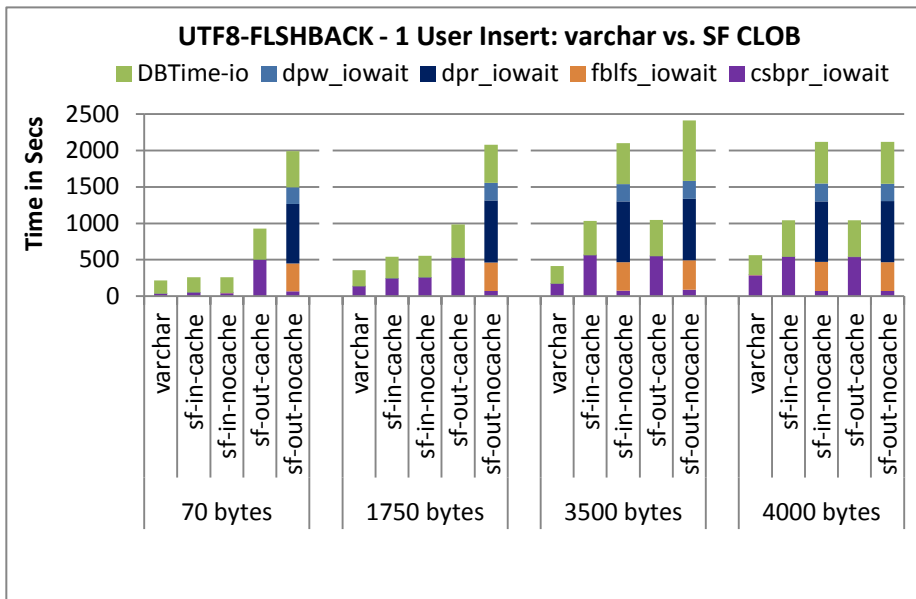
Concurrent Users Select Results

Charts omitted because the results are very similar to UTF8 no archivelog concurrent user selects.

Multi-Byte – UTF-8 + ARCHIVELOG + FLASHBACK

In the following charts we introduce two new statistics: fblfs_iowait = Flashback log file synchronization time; and csbpr_iowait = cell single block physical read time.

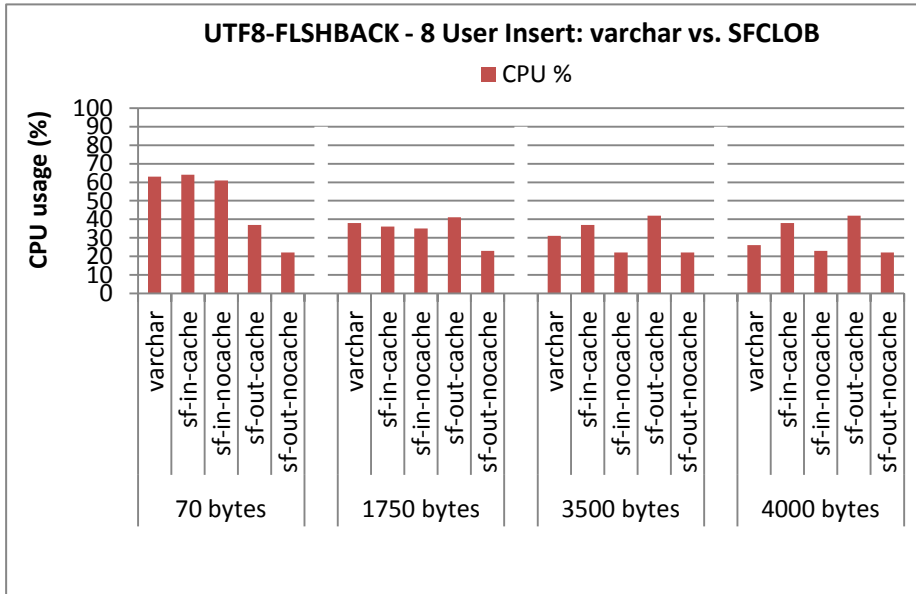
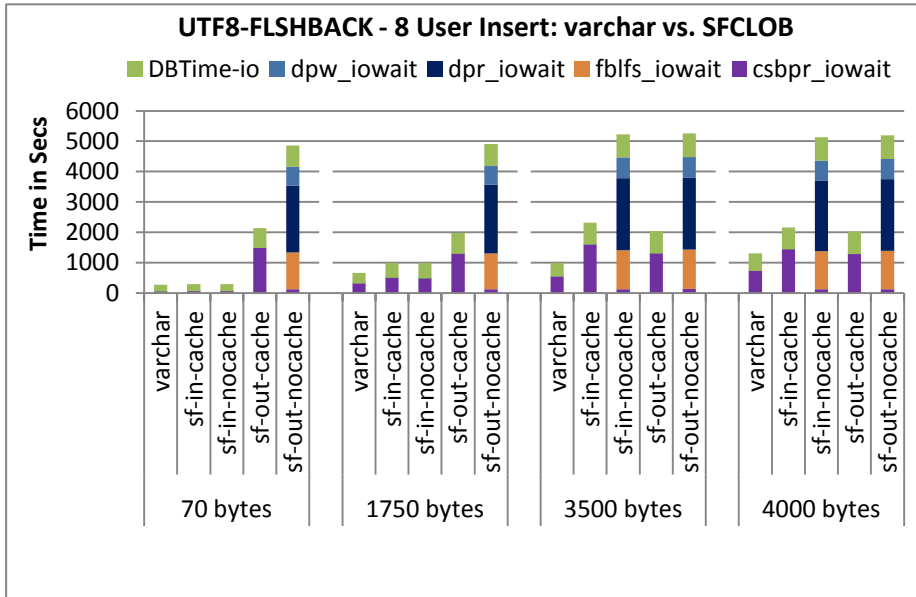
Single User Insert Results



Single User Select Results

Charts omitted because the results are very similar to UTF8 no archivelog single user selects.

Concurrent Users Insert Results



Concurrent Users Select Results

Charts omitted because the results are very similar to UTF8 no archivelog concurrent user selects.

Hardware and Software Details

TABLE 4. HARDWARE AND SOFTWARE DETAILS

COMPONENT	DATABASE NODES	STORAGE CELLS
Number	1	3
CPU	8x Intel Xeon E5430 @ 2.66GHz	8x Intel Xeon E5430 @ 2.66GHz (each cell)
Memory	32 GB	8 GB (each cell)
RDBMS version	12.1.0.1	12.1.1.1
OS: 64-bit Linux	2.6.18 -128.1.16.0.1.el5	2.6.18 -128.1.16.0.1.el5
Disks	---	12 x 300GB, SAS, 15K RPM model HP EF0300FARMU (each cell)

Conclusion

It is not realistic to expect matching performance between VARCHAR and CLOB even in the best case scenario. The differences are justified by the richer functionality the CLOB provides.

Appendix

Configuration: init.ora values

```
lob01.instance_number=1
compatible='12.1.0.0'
control_files='+DATA/LOBDB12c/control1.dbf'
db_block_size=8192
db_files=1024
db_name='LOBDB12c'
log_checkpoints_to_alert='TRUE'
processes=3000
db_cache_size=12G
log_buffer=104857600
shared_pool_size=5G
large_pool_size=512M
pga_aggregate_target=5G
streams_pool_size=128M
java_pool_size=512M
undo_management='AUTO'
lob01.undo_tablespace='UNDO_TS1'
lob02.undo_tablespace='UNDO_TS2'
log_archive_dest='+RECO'
log_archive_format='%t_%s_%r.arc'
db_recovery_file_dest='+RECO'
db_recovery_file_dest_size=1000G
_resource_manager_always_off=true
```

Graph Data

T1 US7ASCII - 1 USER INSERT DB TIME, DIRECT PATH WRITE (DPW) WAIT SUMMARY (ALL TIMES ARE IN SECONDS)

DATASIZE	DATATYPE	WALL ELAPSED	DBTIME (TOTAL)	DPW IOWAIT	CPU%	TAB SIZE (MB)	IDX SIZE (MB)	LOB SIZE
70 bytes	varchar	135	140	1	20	95	40	0
	sf-in-cache	147	152	1	20	130	40	0
	sf-in-nocache	146	150	1	20	130	40	0
	sf-out-cache	300	306	1	30	55	40	8255
	sf-out-nocache	475	479	100	15	55	40	8255
1750 bytes	varchar	155	160	1	20	1985	40	0
	sf-in-cache	168	173	1	20	1985	40	0
	sf-in-nocache	167	172	1	20	1985	40	0
	sf-out-cache	318	323	1	30	55	40	8255
	sf-out-nocache	474	478	100	15	55	40	8255
3500 bytes	varchar	169	174	1	20	3970	40	0
	sf-in-cache	176	181	1	20	3970	40	0
	sf-in-nocache	176	181	1	20	3970	40	0
	sf-out-cache	313	318	1	30	55	40	8255
	sf-out-nocache	477	481	100	15	55	40	8255
4000 bytes	varchar	182	187	1	20	7870	40	0
	sf-in-cache	301	306	1	30	55	40	8255
	sf-in-nocache	478	483	100	15	55	40	8255
	sf-out-cache	320	325	1	30	55	40	8255
	sf-out-nocache	476	480	100	15	55	40	8255

T2 US7ASCII - 1 USER SELECT DB TIME, DIRECT PATH READ (DPR) WAIT SUMMARY (ALL TIMES ARE IN SECONDS)

DATASIZE	DATATYPE	WALL ELAPSED	DBTIME (TOTAL)	DPW IOWAIT	CPU%	TAB SIZE (MB)	IDX SIZE (MB)	LOB SIZE
70 bytes	varchar	35	35	1	15	95	40	0
	sf-in-cache	40	40	1	15	130	40	0
	sf-in-nocache	39	40	1	15	130	40	0
	sf-out-cache	64	65	1	15	55	40	8255
	sf-out-nocache	5030	5035	4875	1	55	40	8255
1750 bytes	varchar	37	40	1	15	1985	40	0
	sf-in-cache	41	45	1	15	1985	40	0
	sf-in-nocache	41	45	1	15	1985	40	0
	sf-out-cache	64	65	1	15	55	40	8255

	sf-out-nocache	4968	4970	4815	1	55	40	8255
3500 bytes	varchar	38	40	1	15	3970	40	0
	sf-in-cache	43	45	1	15	3970	40	0
	sf-in-nocache	43	45	1	15	3970	40	0
	sf-out-cache	66	70	1	15	55	40	8255
	sf-out-nocache	5014	5015	4855	2	55	40	8255
4000 bytes	varchar	39	40	1	15	7870	40	0
	sf-in-cache	65	65	1	15	55	40	8255
	sf-in-nocache	5075	5080	4915	2	55	40	8255
	sf-out-cache	65	65	1	15	55	40	8255
	sf-out-nocache	5053	5055	4895	1	55	40	8255

T3 US7ASCII - 8 USER INSERT DB TIME, DIRECT PATH WRITE (DPW) WAIT SUMMARY (ALL TIMES ARE IN SECONDS)

DATASIZE	DATATYPE	WALL ELAPSED	DBTIME (TOTAL)	DPW IOWAIT	CPU%	TAB SIZE (MB)	IDX SIZE (MB)	LOB SIZE
70 bytes	varchar	22	185	1	71	104	38	0
	sf-in-cache	23	189	1	73	144	38	0
	sf-in-nocache	23	185	1	73	128	38	0
	sf-out-cache	62	498	1	80	55	38	8256
	sf-out-nocache	105	849	368	65	57	38	8256
1750 bytes	varchar	25	204	1	75	2176	38	0
	sf-in-cache	26	213	1	75	1984	38	0
	sf-in-nocache	26	216	1	75	2240	38	0
	sf-out-cache	50	406	1	71	54	38	8256
	sf-out-nocache	105	846	360	66	60	38	8256
3500 bytes	varchar	29	236	1	74	4160	38	0
	sf-in-cache	29	238	1	76	4096	38	0
	sf-in-nocache	29	236	1	77	3968	38	0
	sf-out-cache	62	502	1	80	57	38	8256
	sf-out-nocache	106	857	367	66	57	38	8256
4000 bytes	varchar	31	251	1	76	8000	38	0
	sf-in-cache	63	508	1	79	54	38	8128
	sf-in-nocache	106	854	367	65	56	38	8256
	sf-out-cache	64	521	1	79	57	38	8256
	sf-out-nocache	107	861	366	66	55	38	8256

T4 US7ASCII – 8 USER SELECT DB TIME, DIRECT PATH (DP) READ WAIT SUMMARY(ALL TIMES ARE IN SECONDS)

DATASIZE	DATATYPE	WALL ELAPSED	DBTIME (TOTAL)	DPW IOWAIT	CPU%	TAB SIZE (MB)	IDX SIZE (MB)	LOB SIZE
70 bytes	varchar	6	48	1	68	104	38	0
	sf-in-cache	6	48	1	71	144	38	0
	sf-in-nocache	6	48	1	72	128	38	0
	sf-out-cache	9	78	1	79	55	38	8256
	sf-out-nocache	472	3780	3611	7	57	38	8256
1750 bytes	varchar	6	50	1	73	2176	38	0
	sf-in-cache	6	52	1	74	1984	38	0
	sf-in-nocache	6	52	1	74	2240	38	0
	sf-out-cache	9	78	1	80	54	38	8256
	sf-out-nocache	480	3845	3675	6	60	38	8256
3500 bytes	varchar	6	51	1	71	4160	38	0
	sf-in-cache	7	55	1	73	4096	38	0
	sf-in-nocache	6	55	1	72	3968	38	0
	sf-out-cache	10	80	1	79	57	38	8256
	sf-out-nocache	478	3828	3657	7	57	38	8256
4000 bytes	varchar	6	53	1	73	8000	38	0
	sf-in-cache	10	79	1	81	54	38	8128
	sf-in-nocache	474	3793	3622	7	56	38	8256
	sf-out-cache	10	80	1	80	57	38	8256
	sf-out-nocache	477	3818	3648	7	55	38	8256

T5 UTF8 - 1 USER INSERT DB TIME, DIRECT PATH WRITE (DPW) WAIT SUMMARY(ALL TIMES ARE IN SECONDS)

DATASIZE	DATATYPE	WALL ELAPSED	DBTIME (TOTAL)	DPW IOWAIT	CPU%	TAB SIZE (MB)	IDX SIZE (MB)	LOB SIZE
70 bytes	varchar	129	134	1	20	96	38	0
	sf-in-cache	141	145	1	21	200	38	0
	sf-in-nocache	137	140	1	20	200	38	0
	sf-out-cache	300	303	1	29	53	38	8256
	sf-out-nocache	481	484	93	15	53	38	8256
1750 bytes	varchar	146	150	1	19	1984	38	0
	sf-in-cache	192	196	1	20	3968	38	0
	sf-in-nocache	188	192	1	19	3968	38	0
	sf-out-cache	333	336	1	30	54	38	8256
	sf-out-nocache	510	513	89	15	54	38	8256

3500 bytes	varchar	158	177	1	19	3968	38	0
	sf-in-cache	349	354	1	29	54	38	8256
	sf-in-nocache	562	567	90	14	54	38	8256
	sf-out-cache	364	369	1	30	54	38	8256
	sf-out-nocache	540	545	89	14	54	38	8256
4000 bytes	varchar	200	205	1	20	7872	38	0
	sf-in-cache	359	363	1	30	54	38	8256
	sf-in-nocache	565	569	88	14	54	38	8256
	sf-out-cache	368	373	1	30	54	38	8256
	sf-out-nocache	571	576	89	14	54	38	8256

T5 UTF8 - 1 USER INSERT DB TIME, DIRECT PATH WRITE (DPW) WAIT SUMMARY (ALL TIMES ARE IN SECONDS)

DATASIZE	DATATYPE	WALL ELAPSED	DBTIME (TOTAL)	DPW IOWAIT	CPU%	TAB SIZE (MB)	IDX SIZE (MB)	LOB SIZE
70 bytes	varchar	129	134	1	20	96	38	0
	sf-in-cache	141	145	1	21	200	38	0
	sf-in-nocache	137	140	1	20	200	38	0
	sf-out-cache	300	303	1	29	53	38	8256
	sf-out-nocache	481	484	93	15	53	38	8256
1750 bytes	varchar	146	150	1	19	1984	38	0
	sf-in-cache	192	196	1	20	3968	38	0
	sf-in-nocache	188	192	1	19	3968	38	0
	sf-out-cache	333	336	1	30	54	38	8256
	sf-out-nocache	510	513	89	15	54	38	8256
3500 bytes	varchar	158	177	1	19	3968	38	0
	sf-in-cache	349	354	1	29	54	38	8256
	sf-in-nocache	562	567	90	14	54	38	8256
	sf-out-cache	364	369	1	30	54	38	8256
	sf-out-nocache	540	545	89	14	54	38	8256
4000 bytes	varchar	200	205	1	20	7872	38	0
	sf-in-cache	359	363	1	30	54	38	8256
	sf-in-nocache	565	569	88	14	54	38	8256
	sf-out-cache	368	373	1	30	54	38	8256
	sf-out-nocache	571	576	89	14	54	38	8256

T6 UTF8 - 1 USER SELECT DB TIME, DIRECT PATH READ (DPR) WAIT SUMMARY (ALL TIMES ARE IN SECONDS)

DATASIZE	DATATYPE	WALL ELAPSED	DBTIME (TOTAL)	DPW IOWAIT	CPU%	TAB SIZE (MB)	IDX SIZE (MB)	LOB SIZE
70 bytes	varchar	35	36	1	13	95	40	0
	sf-in-cache	40	41	1	13	130	40	0
	sf-in-nocache	41	42	1	13	130	40	0
	sf-out-cache	65	66	1	14	55	40	8255
	sf-out-nocache	5070	5072	4909	2	55	40	8255
1750 bytes	varchar	37	38	1	10	1985	40	0
	sf-in-cache	64	65	1	14	1985	40	0
	sf-in-nocache	68	69	1	13	1985	40	0
	sf-out-cache	86	88	1	14	55	40	8255
	sf-out-nocache	5059	5061	4873	1	55	40	8255
3500 bytes	varchar	38	40	1	13	3970	40	0
	sf-in-cache	108	110	1	14	3970	40	0
	sf-in-nocache	5070	5072	4859	2	3970	40	0
	sf-out-cache	109	110	1	15	55	40	8255
	sf-out-nocache	5101	5103	4891	2	55	40	8255
4000 bytes	varchar	39	40	1	14	7870	40	0
	sf-in-cache	114	115	1	14	55	40	8255
	sf-in-nocache	5107	5109	4888	2	55	40	8255
	sf-out-cache	114	115	1	12	55	40	8255
	sf-out-nocache	5101	5104	4882	2	55	40	8255

T7 UTF8 - 8 USER INSERT DB TIME, DIRECT PATH WRITE (DPW) WAIT SUMMARY (ALL TIMES ARE IN SECONDS)

DATASIZE	DATATYPE	WALL ELAPSED	DBTIME (TOTAL)	DPW IOWAIT	CPU%	TAB SIZE (MB)	IDX SIZE (MB)	LOB SIZE
70 bytes	varchar	22	184	1	76	128	38	0
	sf-in-cache	23	190	1	77	232	38	0
	sf-in-nocache	24	196	1	75	200	38	0
	sf-out-cache	62	504	1	83	56	38	8256
	sf-out-nocache	106	854	371	68	55	38	8256
1750 bytes	varchar	25	205	1	79	2112	38	0
	sf-in-cache	32	259	1	80	3968	38	0
	sf-in-nocache	31	256	1	81	4096	38	0
	sf-out-cache	65	522	1	82	58	38	8256

	sf-out-nocache	108	865	355	63	55	38	8256
3500 bytes	varchar	28	230	1	77	4096	38	0
	sf-in-cache	68	550	1	82	57	38	8256
	sf-in-nocache	112	903	365	68	56	38	8256
	sf-out-cache	60	487	1	84	56	38	8259
	sf-out-nocache	112	903	364	68	55	38	8256
4000 bytes	varchar	31	256	1	79	8064	38	0
	sf-in-cache	68	550	1	82	59	38	8256
	sf-in-nocache	114	919	374	71	56	38	8256
	sf-out-cache	69	558	1	83	54	38	8256
	sf-out-nocache	113	905	362	69	57	38	8512

T8 UTF8 – 8 USER SELECT DB TIME, DIRECT PATH READ (DPR) WAIT SUMMARY (ALL TIMES ARE IN SECONDS)

DATASIZE	DATATYPE	WALL ELAPSED	DBTIME (TOTAL)	DPW IOWAIT	CPU%	TAB SIZE (MB)	IDX SIZE (MB)	LOB SIZE
70 bytes	varchar	5	44	1	73	104	38	0
	sf-in-cache	6	51	1	75	144	38	0
	sf-in-nocache	6	51	1	73	128	38	0
	sf-out-cache	10	79	1	82	55	38	8256
	sf-out-nocache	491	3936	3764	7	57	38	8256
1750 bytes	varchar	6	47	1	75	2176	38	0
	sf-in-cache	9	75	1	82	1984	38	0
	sf-in-nocache	9	74	1	82	2240	38	0
	sf-out-cache	12	101	1	85	54	38	8256
	sf-out-nocache	497	3978	3779	7	60	38	8256
3500 bytes	varchar	6	50	1	74	4160	38	0
	sf-in-cache	15	124	1	79	4096	38	0
	sf-in-nocache	503	4024	3795	8	3968	38	0
	sf-out-cache	15	124	1	86	57	38	8256
	sf-out-nocache	504	4037	3810	8	57	38	8256
4000 bytes	varchar	6	52	1	75	8000	38	0
	sf-in-cache	16	129	1	83	54	38	8128
	sf-in-nocache	509	4077	3837	11	56	38	8256
	sf-out-cache	16	132	1	86	57	38	8256
	sf-out-nocache	506	4053	3818	8	55	38	8256

T9 UTF8+FLASHBACK – 1 USER INSERT DB TIME, DIRECT PATH WRITE (DPW), DIRECT PATH READ (DPR), FLASHBACK LOG FILE SYNC (FBLFS) AND CELL SINGLE BLOCK PHYSICAL READ (CSBPR) WAIT SUMMARY (ALL TIMES ARE IN SECONDS)

DATASIZE	DATATYPE	WALL ELAPSED	DBTIME (TOTAL)	DPW IOWAIT	DPR IOWAIT	FBLFS IOWAIT	CSBPR IOWAIT	CPU %
70 bytes	varchar	215	218	1	1	1	32	15
	sf-in-cache	257	262	1	1	1	49	14
	sf-in-nocache	255	260	1	1	1	42	14
	sf-out-cache	925	930	1	1	1	500	14
	sf-out-nocache	1984	1990	228	819	380	67	7
1750 bytes	varchar	354	358	1	1	1	135	12
	sf-in-cache	535	541	1	1	1	242	11
	sf-in-nocache	549	556	1	1	1	255	11
	sf-out-cache	981	986	1	1	1	525	13
	sf-out-nocache	2074	2079	240	852	393	70	7
3500 bytes	varchar	409	413	1	1	1	170	12
	sf-in-cache	1030	1035	1	1	1	562	12
	sf-in-nocache	2095	2100	240	833	394	74	7
	sf-out-cache	1038	1048	1	1	1	548	14
	sf-out-nocache	2125	2413	246	847	401	90	7
4000 bytes	varchar	559	564	1	1	1	288	10
	sf-in-cache	1035	1040	1	1	1	539	14
	sf-in-nocache	2116	2120	245	832	398	73	7
	sf-out-cache	1037	1043	1	1	1	537	14
	sf-out-nocache	2110	2118	239	841	393	72	7

Table, Index and Lob segment size is the same as that of UTF8 test case (refer to table T5).

T10 UTF8+FLASHBACK - 8 USER INSERT DB TIME, DIRECT PATH WRITE (DPW), DIRECT PATH READ (DPR), FLASHBACK LOG FILE SYNC (FBLFS) AND CELL SINGLE BLOCK PHYSICAL READ (CSBPR) WAIT SUMMARY (ALL TIMES ARE IN SECONDS)

DATASIZE	DATATYPE	WALL ELAPSED	DBTIME (TOTAL)	DPW IOWAIT	DPR IOWAIT	FBLFS IOWAIT	CSBPR IOWAIT	CPU %
70 bytes	varchar	35	267	1	1	1	50	63
	sf-in-cache	48	297	1	1	1	61	64
	sf-in-nocache	47	297	1	1	1	61	61
	sf-out-cache	1234	2132	1	1	1	1482	37
	sf-out-nocache	1062	4858	627	2198	1214	124	22
1750 bytes	varchar	297	658	1	1	1	318	38
	sf-in-cache	568	993	1	1	1	501	36
	sf-in-nocache	566	987	1	1	1	484	35

	sf-out-cache	1211	1974	1	1	1	1292	41
	sf-out-nocache	1061	4914	613	2270	1181	119	23
3500 bytes	varchar	580	989	1	1	1	538	31
	sf-in-cache	1248	2318	1	1	1	1594	37
	sf-in-nocache	1062	5230	679	2371	1286	126	22
	sf-out-cache	1212	2042	1	1	1	1301	42
	sf-out-nocache	1087	5254	686	2368	1292	135	22
4000 bytes	varchar	1082	1306	1	1	1	721	26
	sf-in-cache	1229	2163	1	1	1	1443	38
	sf-in-nocache	1063	5131	658	2326	1250	128	23
	sf-out-cache	1210	2020	1	1	1	1284	42
	sf-out-nocache	1063	5196	664	2366	1260	126	22

Table, Index and Lob segment size is the same as that of UTF8 test case (refer to table T7).



White Paper Title
December 2013

Author: Vinayagam Djegaradjane

Contributing Authors: Krishna Kunchithapadam
and Kevin Jernigan

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

oracle.com



Oracle is committed to developing practices and products that help protect the environment

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 0113

Hardware and Software, Engineered to Work Together