An Oracle White Paper
July 2010

Oracle Berkeley Database 11gR2 Performance Overview

Executive Overview

Performance is often the first critical factor when selecting a database. This white paper presents performance measurements that are intended to help you understand what to expect from Berkeley DB in some common configurations. Your application performance also depends on your data, data access patterns, cache size, other configuration parameters, operating system and hardware. Benchmarks are almost never representative of the performance of a particular application. They do, however, provide guidelines and help to set basic operational expectations.

Introduction

This paper presents information on the throughput of Oracle Berkeley DB (BDB) 11gR2 (11.2.5.0.21) based on tests conducted using certain configurations. Tests include throughput of BDB's key/value API under various configurations and an examination of BDB's SQL implementation using the Wisconsin and TPC-B-like benchmarks.

Test Environment

All tests are run on the following hardware configuration.

TABLE 1 - HARDWARE CONFIGURATION

PROCESSOR	OPERATING SYSTEM	RAM	STORAGE AND SPEED	FILE SYSTEM
Intel Core 2 Duo E8400, 3.00GHz	Red Hat Enterprise Linux Server 5.4	4 GB	SATA, 7200 RPM	EXT3

Key/Value API Performance Overview

Berkeley DB is highly configurable; it is possible to enable or disable major characteristics of database operation such as write-ahead transaction logging or locking for concurrency control. For the purpose of this set of tests, Berkeley DB is examined in two specific configurations.

First, the tests are preformed on the Data Store (DS) feature set, which is one of the configuration options of Berkeley DB. DS is essentially a simple, single threaded, non-transactional storage system. Next, the tests are performed on the Transactional Data Store (TDS) feature set, which is the configuration option of Berkeley DB that provides full transactional semantics.

TABLE 2 - DATA STORE AND TRANSACTIONAL DATA STORE FEATURES

FEATURES	DATA STORE(DS)	TRANSACTIONAL DATA STORE(TDS)
Access Method	Btree	Btree
Locking	None	Page-level
Logging	None	On-disk with 128 MB buffer
Transactions	None	Synchronous
Shared Memory	DB_PRIVATE	DB_PRIVATE
Buffer Cache	512 MB	512 MB

Both set of tests use a 512 MB buffer cache, DB_PRIVATE flag for database environment, and a spatial locality of 10. The DB_PRIVATE flag indicates the use of heap memory for the database cache; this configuration is only suitable for single-process (but potentially multi-threaded) execution.

The most common measure of database performance is *throughput* (measured as records read or written in a fixed time interval). These tests express throughput as operations per second using Berkeley DB 11gR2 (11.2.5.0.21).

The test database uses fixed size records with 64-byte key and 64-byte data values. All key and data values are integers. Each test performs 57,600 batches of ten sequential key operations. For example, on an insert test, a key is randomly generated and inserted along with the next nine consecutive key values. This process is then repeated 57,600 times. Each test is run five times and both the mean and standard deviation for each operation type are reported.

For each configuration, here is the sequence of tests:

- Step 1. Populate the database with 576,000 records (57,600 sets of ten sequential keys).

  This creates the test database and warms the cache, but this time is not reported here.

- Step 2. Retrieve 57,600 random sets of ten sequential keys.
- Step 3. Update 57,600 random sets of ten sequential keys.
- Step 4. Delete all records in sets of ten sequential keys.
- Step 5. Repopulate the database as done in Step 1.

For each type of operation, the timer starts before the first operation and ends after the last operation. The time taken to open or close the environment and database handles are not included.

Data Store: Single Threaded

The first test measures throughput for a single threaded application using Berkeley DB Data Store (DS). The following table shows the results.

TABLE 3 - BDB DS SINGLE THREADED PERFORMANCE

DESCRIPTION	INSERT		FETCH		DELETE		UPDATE	
	OPS/SEC	STD DEV	OPS/SEC	STD DEV	OPS/SEC	STD DEV	OPS/SEC	STD DEV
Data Store	208,139	329	264,665	229	158,506	236	250,297	870

Transactional Data Store: Single Threaded

The second test measures throughput for a single threaded application using Berkeley DB TDS by varying configuration options such as write durability, log durability, and storage medium. Each of these configuration options are explained as follows:

- **Write durability** - By default, transactions are committed synchronously to disk. The DB_TXN_NOSYNC flag enables asynchronous commits, and the DB_TXN_WRITE_NOSYNC flag writes data to the file system but does not perform a synchronous disk write.
- **Log durability** - By default, logs are stored on disk; non-persistent, in-memory logging is also examined.
- **Storage medium differences** - The default configuration uses a traditional hard disk. This performance is compared to that obtained using a 2 GB RAM disk and by storing both the database and logs on the RAM disk. In all cases, transactions are committed synchronously.

TABLE 4 - BDB TDS SINGLE THREADED PERFORMANCE

DESCRIPTION	INSERT		FETCH		DELETE		UPDATE	
	OPS/SEC	STD DEV	OPS/SEC	STD DEV	OPS/SEC	STD DEV	OPS/SEC	STD DEV
TDS SYNC	693	10	159,837	895	831	23	1,732	18
TDS RAMDISK[1]	49,673	912	162,848	1,588	44,279	277	60,973	291
TDS WNS[2]	37,664	460	160,307	293	36,110	454	52,922	258
TDS NS[3]	53,199	613	159,980	1,419	49,340	183	86,960	639
TDS INMEM[4]	66,435	102	163,229	815	58,845	101	97,602	396

[1] 2GB RAMDISK, database and logs are stored on RAMDISK

[2] DB_TXN_WRITE_NOSYNC transaction commit flag

[3] DB_TXN_NOSYNC transaction commit flag

[4] In-memory logging enabled

Because the database cache is sufficient for all the tests above, disk I/O does not factor into the read throughput. By comparing results for TDS and TDS RAMDISK, it is clear that I/O latency and throughput can greatly affect the throughput for writes. The data above also shows the impact of durability on write throughput as shown by the difference between TDS SYNC, TDS INMEM, TDS WNS, and TDS NS.

The results show a number of interesting characteristics common to all databases. One such characteristic is that the I/O operation is the most critical factor in determining performance. While reads from storage disk do incur a small penalty, writes are, by far, the critical path. In the TDS insert and update tests the greatest impact is on performance, because the tests incur the most I/O overhead. These tests show you the five major categories of latency:

- memory and processor
- user to kernel space transfer
- file system
- storage I/O
- media

TDS INMEM is the fastest because all log operations are in memory. Commit operations are performed in the buffer cache, with no data transfer to either the kernel file system buffers or disk. The TDS INMEM single threaded test measures of insert operations shows the ideal speed for your system, but may not be very useful because data is not permanent. When the process terminates unexpectedly the data is deleted. As the data was in memory there is no durability.

When a transaction is configured with the DB_TXN_NOSYNC flag, no data is transferred from log buffers to the file system on commit. Instead, log data stays in the log buffer until the buffer fills, at which point a file system write transfers the log records from user memory to kernel/file system memory. The file system then determines when, if ever, that data is written to the storage media. Thus, the performance difference between TDS INMEM and TDS NS reflects the overhead caused by the transfer of log records from the user-level log buffer into the kernel/file system memory, triggered when the log buffer fills.

When a transaction is configured with the DB_TXN_WRITE_NOSYNC flag, Berkeley DB transfers log records from its user-level log buffer memory into the operating system/file system at each transaction. This provides durability as long as the operating system does not fail before writing that data to stable storage. In the case of application failure, no data is lost if the operating system continues to run. However, if the operating system or hardware fails, then the data transferred to the operating system but not yet written to the stable media, is lost.

The performance difference between TDS WNS and TD SNS during the insert operation reflects these commit-time copies from the Berkeley DB log buffer to the file system. Finally, by comparing the performance of TDS SYNC to TDS RAMDISK, you can observe the difference between memory writes and disk writes. In the case of TDS SYNC, the overhead of the SATA bus hardware and the hard drive cause a dramatic performance decrease.

It is interesting to note that the read performance across all tests is roughly the same with the average varying only 1% from the standard deviation (sd = 1655.92, avg = 161240.2, percent variation = 1.026989482). This indicates that the cache is identically efficient across tests and that the overhead of transactional guarantees on fetch (read) operations for data in-cache is negligible. Although not measured, the I/O overhead can be expected to dominate the processing time when requesting (fetching, reading) out-of-cache data.

Transactional Data Store: Scaling on Symmetric Multiprocessing Systems (SMPs)

The following table shows how the performance of BDB TDS scales with Symmetric Multiprocessing Systems (SMPs). With multi-threaded access, each thread is given a DB handle to minimize contention. Each thread uses a timer to calculate its throughput. The aggregate throughput is reported in the table.

When a second thread is introduced, throughput drops for each operation type because the system now performs locking and hence introduces overhead. The write throughput scales well with additional threads. Read throughput decreases past four threads because the CPU utilization is 100% and additional threads cause unnecessary context switches and hardware cache eviction.

TABLE 5 - TDS PERFORMANCE FOR MULTIPLE THREADS ON AN SMP SYSTEM

DESCRIPTION	INSERT		FETCH		DELETE		UPDATE	
	OPS/SEC	STD DEV	OPS/SEC	STD DEV	OPS/SEC	STD DEV	OPS/SEC	STD DEV
TDS 1 thread	693	10	159,837	895	831	23	1,732	18
TDS 2 threads	647	7	106,196	432	767	20	1,634	27
TDS 3 threads	690	9	126,762	762	863	20	1,909	32
TDS 4 threads	721	8	134,581	497	882	18	2,085	54
TDS 8 threads	859	26	128,928	423	967	27	2,497	32
TDS 12 threads	930	24	112,735	518	1,013	20	2,658	149
TDS 16 threads	968	13	99,860	820	1,085	34	2,837	117

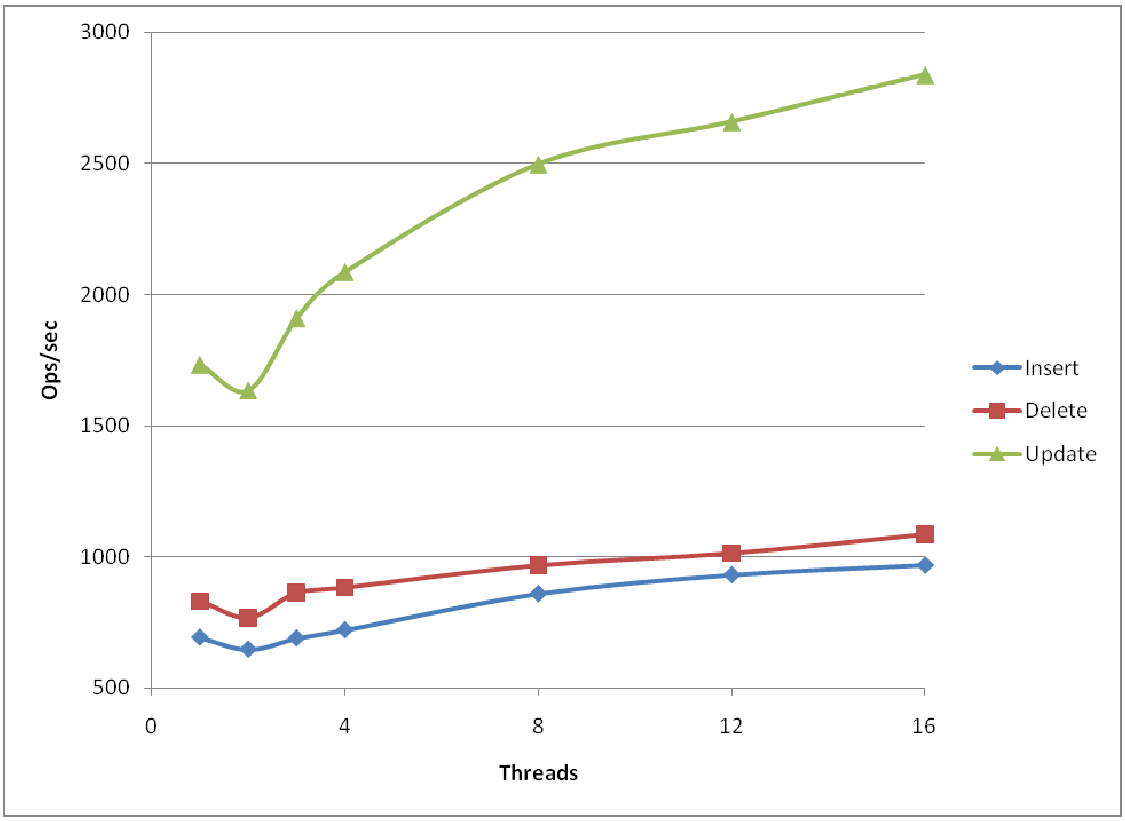

Figure 1. TDS performance for multiple threads on an SMP system

SQL Interface Performance Overview

Several popular benchmarks are specified for SQL-based relational databases. Berkeley DB's SQL interface lends itself to adaptations of these benchmarks.

These tests are inspired by two popular benchmarks: the Transaction Processing Performance Council's TPC Benchmark B[1], and D. J. DeWitt's Wisconsin benchmark[2].

These implementations and test runs are not certified nor reviewed by any third party; they represent our attempt to develop a rough understanding of BDB's performance characteristics, without reference to those of other software systems.

TPC Benchmark B

The TPC-B-like test implements the schema and standard transaction as specified by the TPC Benchmark B specification[3]. The number of accounts is scaled down to make it manageable to run; it has 1000 branches with 10,000 tellers and 100,000 accounts.

The following table shows that TPC Benchmark B produces a single number representing the transaction throughput rate in transactions per second (TPS). These results are obtained using a 256 MB database cache.

[1] http://www.tpc.org/tpcb/default.asp

[2] http://firebird.sourceforge.net/download/test/wisconsin_benchmark_chapter4.pdf

[3] http://www.tpc.org/tpcb/spec/tpcb_current.pdf

TABLE 6 - TPC-B TRANSACTION THROUGHPUT RATE

THREADS	TPS
1	1846.71
2	2310.84
3	2508.26
4	2678.14
5	2808.51
10	2859.15

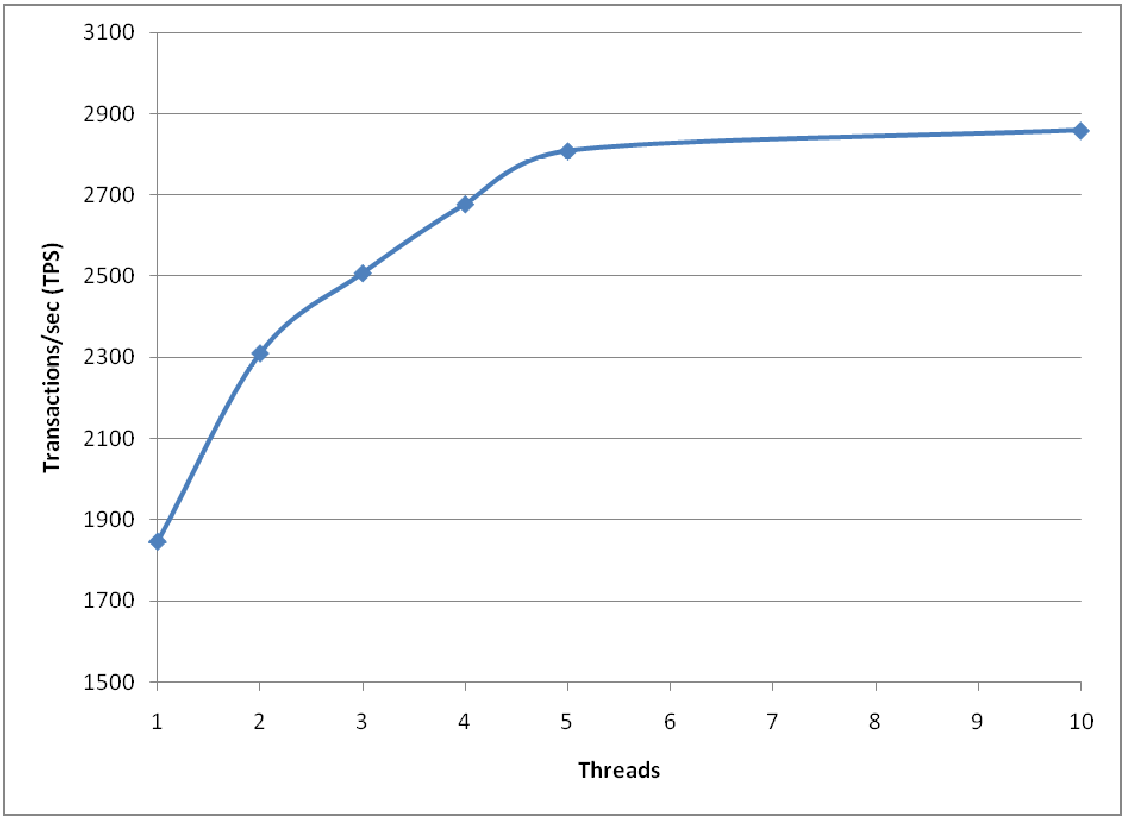

Figure 2. TPC-B-like benchmark measuring transactions per second (TPS) versus number of worker threads

Wisconsin

The performance test attempts to adhere to the schema and queries described in the Wisconsin benchmark specification[4], but there has been no attempt to mitigate effects of caching in the database system and the operating system.

This test measures the single-user performance of a large number of different test cases on the standard database. The composite result for all test cases can be used to analyze the strengths and weaknesses of a database. For the descriptions of the queries, see Appendix I in the paper referenced below.

For these test runs, DB is built with compile-time options to optimize for single-threaded and single-user access. Each query is run ten times, and the average elapsed time is reported in the table below, in milliseconds.

TABLE 7 - WISCONSIN BENCHMARK MEASURING AVERAGE ELAPSED TIME

CASE #	ELAPSED (MS)	CASE #	ELAPSED (MS)	CASE #	ELAPSED (MS)	CASE #	ELAPSED (MS)
1	8.175	9	31047.220	17	16.944	25	18.163
2	13.875	10	3160.824	18	58.801	26	0.907
3	0.879	11	5335.219	19	136.595	27	3.844
4	6.749	12	9.920	20	3.944	28	4.380
5	1.368	13	9.942	21	32.566	29	1.081
6	11.017	14	11.585	22	32.774	30	1.273
7	0.244	15	19.407	23	0.102	31	1.252
8	1.373	16	14.759	24	17.742	32	1.081

[4] http://firebird.sourceforge.net/download/test/wisconsin_benchmark_chapter4.pdf

Conclusion

Berkeley DB key/value API can perform well in both single threaded and multi-threaded applications. Also, if the application does not require full durability, it is possible to significantly enhance the performance.

Berkeley DB was enhanced in 11gR2 5.0 with the addition of a SQL API built on the key/value API.

You can download Oracle Berkeley DB at:

http://www.oracle.com/technology/software/products/berkeley-db/index.html

You can post your comments and questions at the Oracle Technology Network (OTN) forum for Oracle Berkeley DB at:

http://forums.oracle.com/forums/forum.jspa?forumID=271

For sales or support information, email to: berkeleydb-info_us@oracle.com

Find out about new product releases by sending an email to: bdb-join@oss.oracle.com

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

Oracle Berkeley Database 11gR2
Performance
July 2010
Author: Karl Fu

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