OLTP BENCHMARK

ARPAN MITTAL
3234016253

OUTLINE

1. Introduction
2. Benhmark
3. Results

Introduction

OLTPBenchmark is a multi-threaded load generator. The framework is designed to be able to produce variable rate, variable mixture load against any JDBC-enabled relational database. The framework also provides data collection features, e.g., per-transaction-type latency and throughput logs.

Testing System Details

Specification of VM:
Standard B4ms (4 vcpus, 16 GB memory)
Cache details: memory 10Gb, **max connections 10**, reserved space 10%

SMALLBANK

SmallBank : This workload models a banking application where transactions perform simple read and update operations on customers accounts. All of the transactions involve a small number of tuples. The transactions' access patterns are skewed such that a small number of accounts receive most of the requests.

Procedure distribution for testing

15% Amalgamate

15% Balance

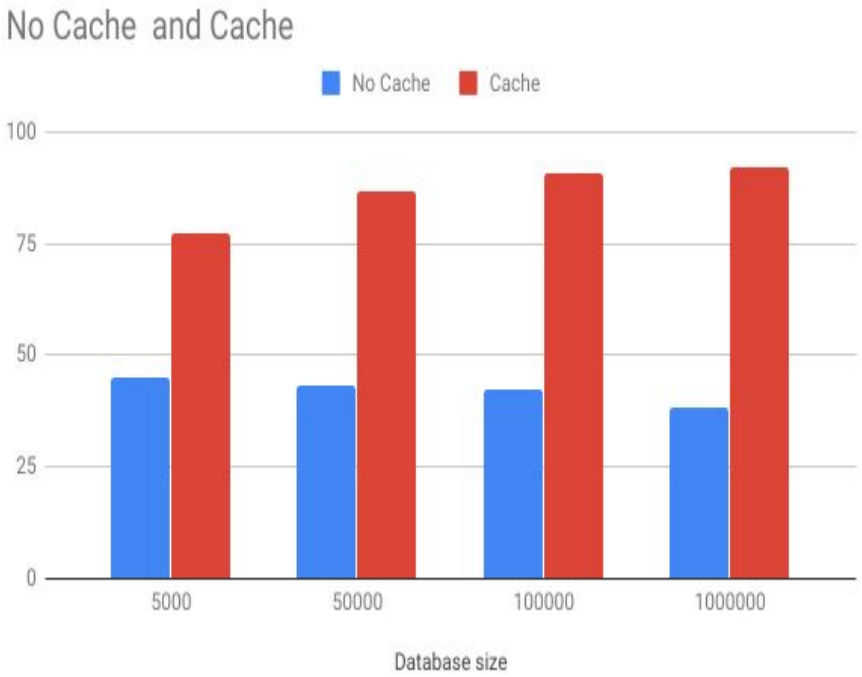
15% Deposit Checking

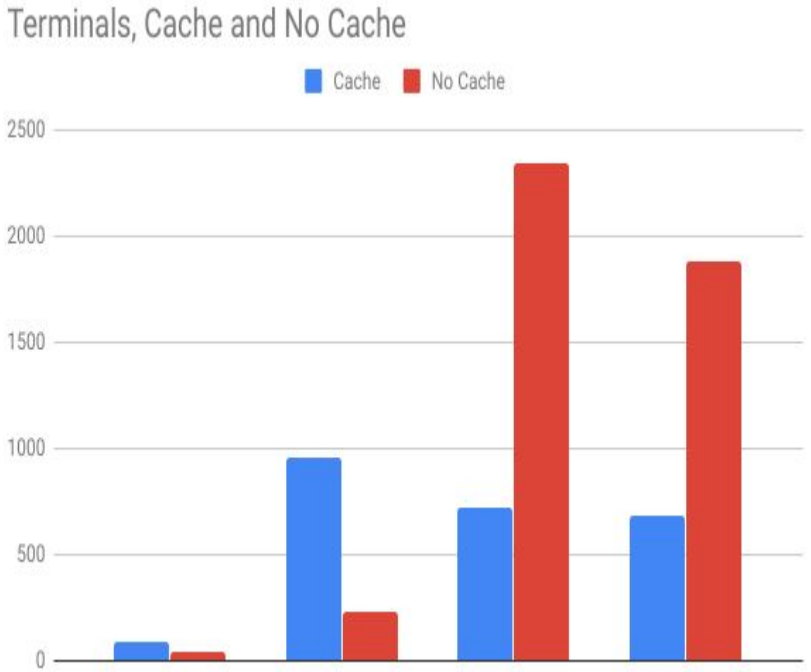
25% Send Payment

15% TransactSavings

15% WriteCheck

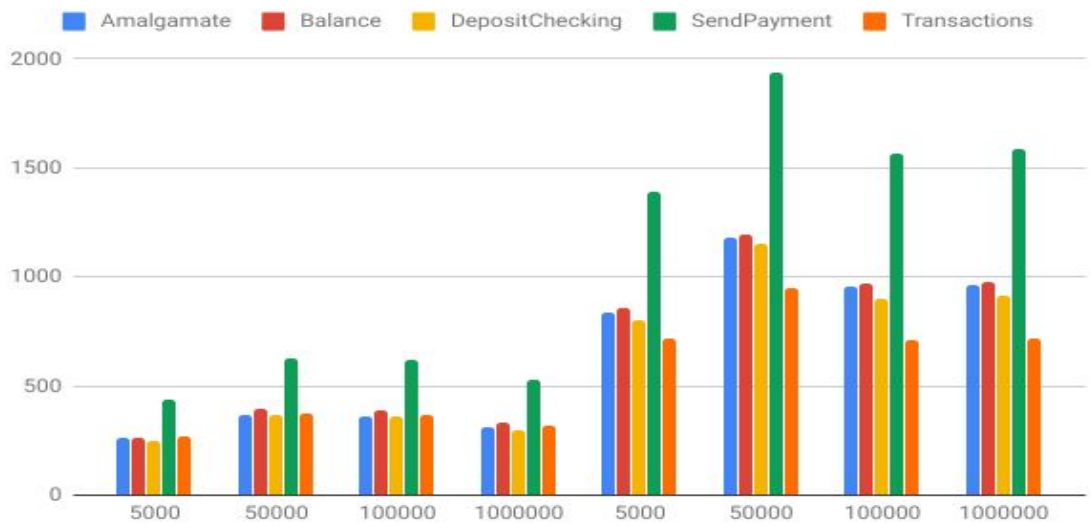
Results(Smallbank)





Results(small bank)...

Number of committed transactions



YCSB

The Yahoo! Cloud Serving Benchmark (YCSB) is a collection of micro-benchmarks that represent data management applications whose workload is simple but requires high scalability [16]. Such applications are often large-scale services created by Web-based companies. Although these services are often deployed using distributed key/value storage systems, this benchmark can also provide insight into the capabilities of traditional DBMSs. The YCSB workload contains various combinations of read/write operations and access distributions that match products inside Yahoo! It is representative of simple key-value store applications. The benchmark has been leveraged in previous studies for exploiting the trade-offs between availability/consistency/partition tolerance, and more generally to showcase storage engines and caching results (e.g., improving the throughput of random writes)

YCSB(Result)

Query distribution:

5% ReadRecord

20% Update

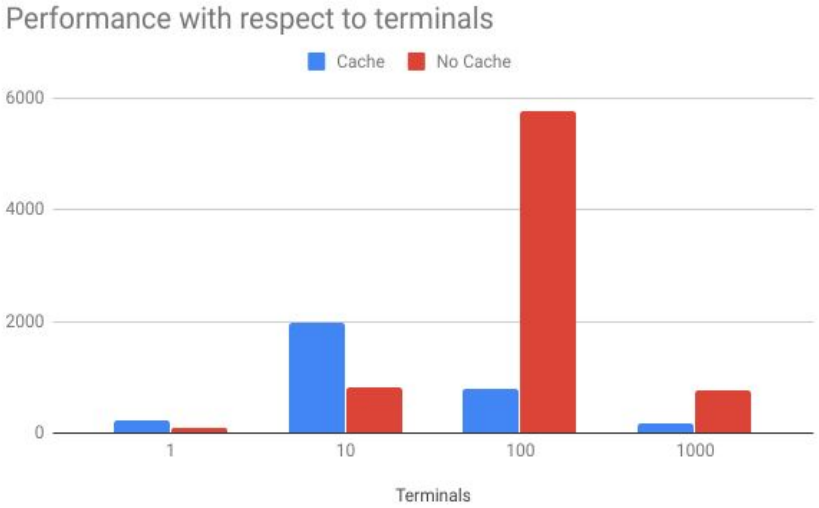
5% Scan

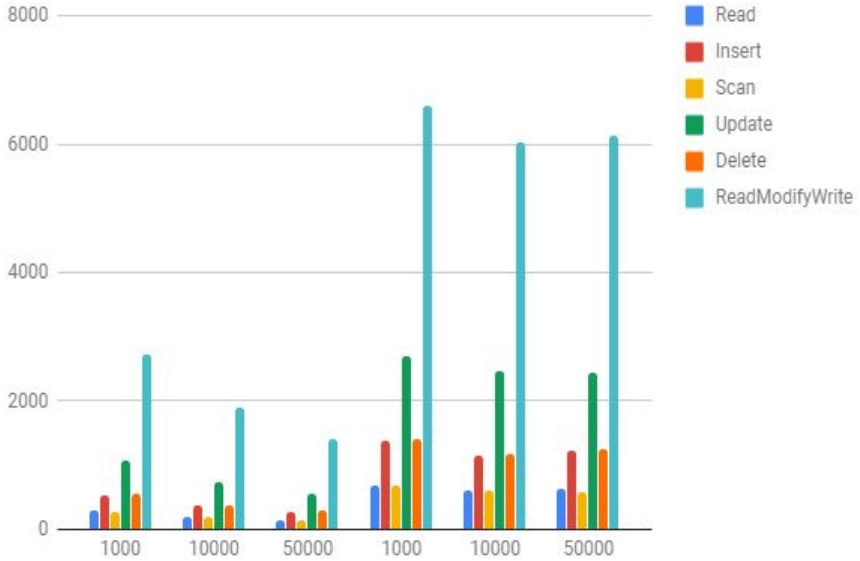
10% Delete

50% RMW(Read Modify write)

10% Insert

Results(YCSB)

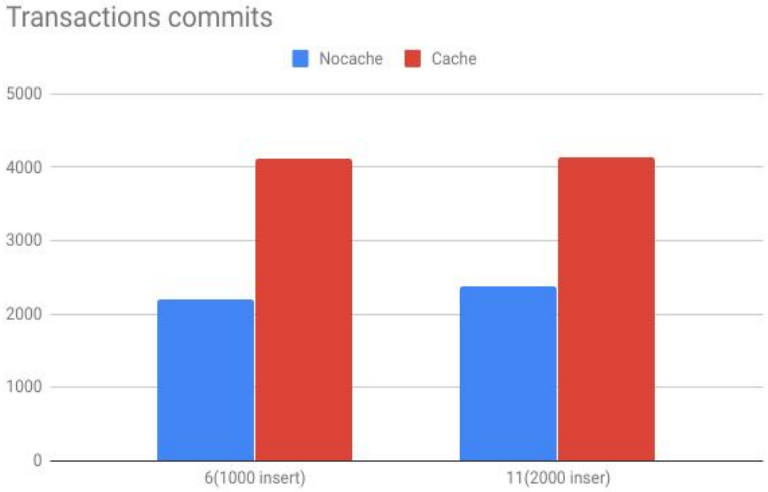




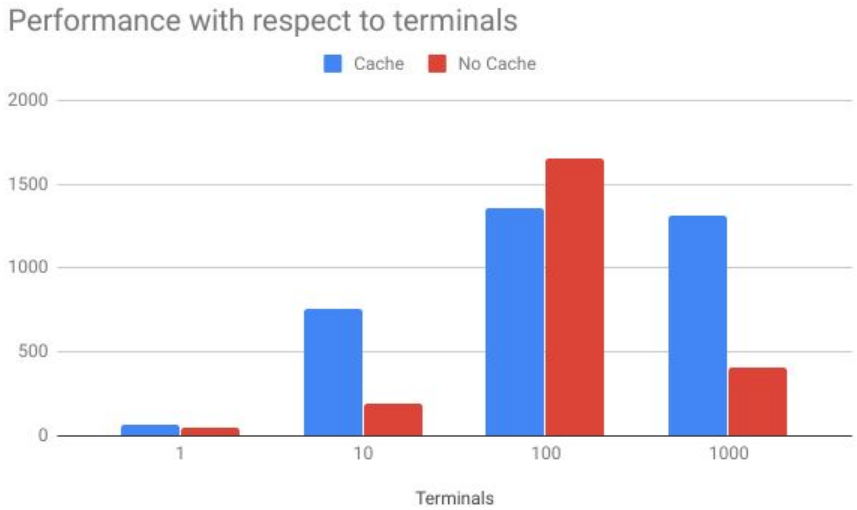
VOTER

The Voter workload is derived from the software system used to record votes for a Japanese and Canadian television talent show. As users call in to vote on their favorite contestant during the show, the application invokes transactions that update the total number of votes for each contestant. The DBMS records the number of votes made by each user up to a fixed limit. A separate transaction is periodically invoked to compute vote totals during the show. This benchmark is designed to saturate the DBMS with many short-lived transactions that all update a small number of records.

Voter(Result)



Voter Result(continue)..



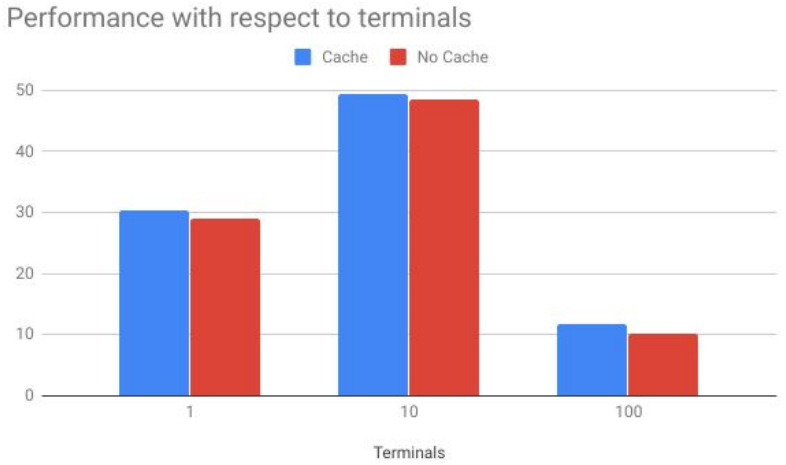
SIBENCH

SIBench is a microbenchamark designed to explore snapshot isolation in DBMSs [12]. It contains a single key/value table and two transactions that fetch the minimum value of a column or increment a single value of an entry. This workload creates a situation where the DBMS must resolve read-write conflicts while also stressing the CPU by scanning the table for the minimum value

Result(SIBench)

Query distribution:
50% Minoperation

50% UpdateOperation



Results(SIBENCH)

Left one are without cache other are with cache