Overview of TPC Benchmark E: The Next Generation of OLTP Benchmarks

Trish Hogan

IBM
3039 Cornwallis Road,
Research Triangle Park
NC 27709
thogan@us.ibm.com

Abstract. Set to replace the aging TPC-C, the TPC Benchmark E is the next generation OLTP benchmark, which more accurately models client database usage. TPC-E addresses the shortcomings of TPC-C. It has a much more complex workload, requires the use of RAID-protected storage, generates much less I/O, and is much cheaper and easier to set up, run, and audit. After a period of overlap, it is expected that TPC-E will become the de facto OLTP benchmark.

Keywords: TPC-E, TPC-C, OLTP, database performance, benchmark.

1 Introduction

TPC Benchmark™ E (TPC-E) is a database server benchmark that measures OLTP performance. The benchmark is one of several that have been produced by the Transaction Processing Performance Council (TPC), a non-profit corporation founded to define transaction processing and database benchmarks and to disseminate objective, verifiable TPC performance data to the industry. The TPC has many member companies such as IBM, Dell, HP, Intel, Oracle, Microsoft and AMD.

The TPC-E benchmark uses a database to model a brokerage firm with customers who generate transactions related to trades, account inquiries, and market research. The brokerage firm in turn interacts with financial markets to execute orders on behalf of the customers and updates relevant account information.

The benchmark is "scalable," meaning that the number of customers defined for the brokerage firm can be varied to represent the workloads of different-size businesses. The benchmark defines the required mix of transactions the benchmark must maintain. The TPC-E metric is given in transactions per second E (tpsE). It specifically refers to the number of Trade-Result transactions the server can sustain over a period of time.

The first section of this paper, "Why do we need a new OLTP benchmark?" compares and contrasts TPC-E and TPC-C, and discusses the benefits of the new benchmark. The second section, "Overview of TPC-E," provides more detailed information about the benchmark, including a description of the business model used

R. Nambiar and M. Poess (Eds.): TPCTC 2009, LNCS 5895, pp. 84–98, 2009.

for the benchmark, the kind of information stored in the database, and the types of transactions performed. The final section, “Transition to TPC-E,” briefly addresses the coexistence of the TPC-C and the TPC-E and the eventual sun setting of the TPC-C benchmark once TPC members begin to publish results for the new benchmark.

The overall goal is to introduce the audience to the new TPC-E benchmark. Some knowledge and understanding of the TPC-C benchmark on the reader’s part is assumed. Another goal is to help the audience understand how to use TPC-E benchmark results to evaluate the performance of database servers.

For more detailed information about the TPC and the TPC-E benchmark, visit the TPC Web site at www.tpc.org.

2 Why Do We Need a New OLTP Benchmark?

Why TPC-E rather than TPC-C? Benchmarks have a life time. Good benchmarks drive industry and technology forward. At some point, all reasonable advances have been made using a particular benchmark. When that happens, benchmarks can become counterproductive by encouraging artificial optimizations. So, even good benchmarks become obsolete over time. The TPC-C Specification was approved July 23, 1992. Since then, it has become the de facto industry-standard OLTP benchmark, but now TPC-C is roughly 17 years old. In “dog years” that’s 119. In “computer years,” it’s basically ancient!

2.1 TPC-C: An Aging Benchmark Losing Relevance

Static Transaction Profiles

The benchmark is running the same workload today as when it was introduced in 1992. TPC-C’s transactions are too “lightweight” by today’s standards, but it is not practical to modify the existing workload because that would break comparability. (The companies that published the benchmark results don’t want to lose their investment by losing comparability.) Those who run the benchmark have a very good understanding of the workload and how to super-tune it.

Unbalanced System Configurations

Over the years, with the advance of technology, we’ve seen that:

- CPU performance has grown according to Moore’s Law.
- Disk drive latencies have not improved substantially.
- Memory has grown disproportionately to I/Os per system.
- The TPC-C workload has not changed.
- Ongoing improvements to software have led to increasingly higher TPC-C scores.

As a result of these factors, running the benchmark requires larger and larger I/O subsystems, which in turn, increases the cost, which is borne entirely by the benchmark sponsor. For example, a recent TPC-C result on an IBM System x3850 M2 used four processors, 256GB of memory, and 1,360 disk drives for the database.

Compare that to the results published in October 2000 for the IBM Netfinity 7600 with four processors, 8GB of memory, and 236 drives.[1]

Today the workload and configurations are less representative of clients' environments than when the benchmark was first introduced. Because of the amount of hardware required, the benchmark has become too expensive and requires too much time to run and audit.

2.2 TPC-E: A Benchmark More Relevant to Today's Application Environment

What is needed and what TPC-E delivers is a new OLTP database-centric benchmark that:

- Provides comparable results (i.e., results from different vendors can be compared)
- Represents a familiar business model that is easy to understand
- Reduces the cost and complexity of running the benchmark
- Enhances the complexity of the database schema
- Encourages database uses that are more representative of client environments

More Realistic Benchmark Configurations

Configurations used to run the new benchmark should be more like actual client configurations. This means that the software and hardware configuration used in the benchmark should be similar or the same as what a client would use. The configuration should not have a large I/O subsystem if clients would not have a similarly large I/O subsystem. Having a benchmark configuration that more closely reflects the client's application environment makes it easier to use benchmark results for capacity planning. Any improvements made to the hardware or software to improve the benchmark result would also benefit the client. This is not always the case now because TPC-C configurations do not reflect typical client configurations. The following table shows that TPC-E configurations more closely resemble client configurations.

Table 1. Comparison of TPC-C and TPC-E

Feature	TPC-C	Client	TPC-E
Drives/core	50 to 150	5 to 10	10 to 50
Database layout	Simple	Complex	Complex
Database transactions	Simple	Complex	Complex
Database constraints	None	Enforced	Enforced
RAID-protected data	No	Yes	Yes

[1] IBM System x3850 M2 with the Intel Xeon Processor X7460 2.66GHz (4 processors/24 cores/24 threads), 684,508 tpmC, $2.58 USD / tpmC, total solution availability of October 31, 2008. Results referenced are current as of May 22, 2009. IBM Netfinity 7600 with Intel Pentium® III Xeon (4 processors/4 cores), 32,377 tpmC, $13.70 USD / tpmC, availability of October 25, 2000 (withdrawn).

2.3 TPC-E and TPC-C: Comparison of Key Features

The following table shows the key characteristics of TPC-C and TPC-E side-by-side.

Table 2. Comparison of TPC-C and TPC-E characteristics

Characteristics	TPC-C	TPC-E
Business model	Wholesale supplier	Brokerage house
Number of database tables	9	33
Number of database columns	92	188
Minimum columns per table	3	2
Maximum columns per table	21	24
Datatype count	4	Many
Primary keys	8	33
Foreign keys	9	50
Tables with foreign keys	7	27
Check constraints	0	22
Referential integrity	No	Yes
Database content	Unrealistic	Realistic
Ease of partitioning	Unrealistically easy	Realistic
Database roundtrips per transaction	One	One or many
Number of transactions	5	10
Number of physical I/Os	3x to 5x	x
Client machines needed	Many	Fewer
RAID requirements	Database log only	Everything
Timed database recovery	No	Yes
TPC-provided code	No	Yes

Different Business Model

The TPC-C business model is that of a wholesale supplier, and the database is organized by Warehouses, Districts and Customers. The TPC-E business model is a brokerage house, and the database is organized by Customers, Accounts and Securities.

RAID Protection

A common complaint about TPC-C is that it does not require RAID protection of the disk subsystem that contains the database. In the real world, clients cannot run their databases without some protection against drive failure. TPC-E requires RAID protection and tests how performance is affected when a drive is rebuilding.

Timed Database Recovery

TPC-E requires test sponsors to report how long it takes after a catastrophic failure to get the database back up and running at 95% of the reported throughput. This requirement means that test sponsors can no longer take shortcuts to improve performance while ignoring reliability.

Richer Database Schema and Transactions

TPC-C has nine database tables and five transactions. TPC-E has 33 tables and 10 transactions. TPC-E has many different data types, many primary keys, foreign keys, check constraints, and has referential integrity. TPC-C was very easy to partition because everything was keyed off of the warehouse ID. TPC-E is not unrealistically easy to partition, reflecting the issues clients see when they try to partition their data.

When clients implement transactions in their applications, they often do many roundtrips to the database; that is, they get some data from the database, send the data back to the application, process the data in the application, and then go to the database again for some more data. In TPC-C, the application has been so tuned that all the processing is done in one roundtrip to the database because all the processing is done in a stored procedure in the database. In TPC-E some transactions have rules that the transaction cannot be implemented as a single stored procedure in the database. The rules force more roundtrips to the database.

TPC-E has stricter database isolation requirements than TPC-C. This means that database vendors will have to work on improving database locking performance. Clients will see this performance improvement when they run their applications.

The database schema and transactions for TPC-E are much richer than those of TPC-C. The richer schema means that there are lots of possible ways to optimize performance. Over the life of the benchmark, test sponsors will find ways to use secondary indexes to improve transaction performance. TPC-C has been around so long that all the SQL tuning has been found and done.

Constraint Checking

When TPC-C came out, database constraint checking was not a standard feature for all Database Management Systems; now it is, so constraint checking is enforced in TPC-E.

Configuration Requirements

The transactions in TPC-E are designed to do more logical fetches and consume more CPU time than the TPC-C transactions. Logical fetches are the number of rows a transaction has to read. The rows may already be in memory or they may be out on disk. If the rows are already in memory, then no physical I/O to the disk is required.

During development, extensive prototyping was done to ensure that the TPC-E transactions used a lot of CPU and did some, but not too much, physical I/O. Too much physical I/O requires many more disk drives to ensure that the processor is never idle waiting for physical I/O. The result of this prototyping is that TPC-E does between three to five times less physical I/O than TPC-C. So even though TPC-E requires the disk drives to be RAID-protected, the benchmark still uses significantly fewer disk drives than a TPC-C configuration for the same server with the same number of processors and memory. This is good news for test sponsors. The disk I/O subsystem for TPC-C is incredibly expensive, which means that some hardware vendors cannot afford to run the TPC-C benchmark. TPC-E will be much less expensive to configure and run, so more hardware vendors will be able to publish results. Clients will benefit because they will be able to compare benchmark results from several vendors before deciding which vendor’s products they want to buy.

Table 3. Comparison of TPC-C and TPC-E configurations

	TPC-C	**TPC-E**
System	HP ProLiant DL370 G6	IBM System x3650 M2
Total price	$678,231	$302,146
Score	631,766 tpmC	798.00 tpsE
Price/performance	$1.08/tpmC	$378.63/tpsE
Availability date	3/30/09	6/30/09
Processors/cores/threads	2/8/16	2/8/16
Database server memory	144GB	96GB
Disks	1210 (RAID-0)	450 (RAID-10)
RTEs	18	0 (N/A)
Clients	8	2
Total systems to setup and tune	27	3

Another factor that makes TPC-E less expensive to run is that it requires fewer front-end systems because it focuses on benchmarking the database server. In contrast, TPC-C uses a Web page-based front-end, which emulates users keying in the Web page, and thinking. Here is a comparison of the TPC-C and TPC-E configurations for two very similar database servers.

More Realistic Database Content

TPC-C uses random numbers to pick string fragments and then concatenates the string fragments. TPC-E is populated with pseudo-real data. The distributions are based on:

- 2000 U.S. and Canada census data[2]
  - Used for generating name, address, gender, etc.
  - Introduces natural data skew
- Actual listings on the NYSE and NASDAQ

The benefits of using pseudo-real data are that TPC-E has realistic-looking data that is compressible and can be used for backup testing. The data is closer match to the kind of data found in clients' databases.

Table 4. Sample data from TPC-C CUSTOMER table

C_FIRST	C_MIDDLE	**C_LAST**	**C_STREET1**
RONpTGcv5ZBZO8Q	OE	BARBARABLE	bR7QLfDBhZPHlyDXs
e8u6FMxFLtt6p Q	OE	BARBARPRI	eEbgKxoIzx99ZTD S
bTUkSuVQGdXLjGe	OE	BARBARPRES	QCGLjWnsqSQPN DS
18AEf3ObueKvubUX	OE	BARBARESE	JnBSg4RtZbALYu S
mFFsJYeYE6AR bUX	OE	BARBARANTI	MLEwwdy3dXfqngFcE

[2] Only names from the 2000 census have been used—all other data are fictional and any similarities are purely coincidental.

Table 5. Sample data from TPC-E CUSTOMER table

C_TAX_ID	C_L_NAME	**C_F_NAME**	C_M_NAME	**C_DOB**
757FI2006HD923	Mexicano	Courtney	T	1997-11-30
922SN3775RQ823	Udley	Judith	F	1954-09-27
006GT3444BE624	Buchanan	John	R	1971-06-13
181UZ4114LR434	Soloman	Clinton	D	1938-02-27
355IE4773VF335	Orner	Harry	P	1974-11-15

Provided Code

TPC does not provide code for the TPC-C benchmark, but it does provide code for the TPC-E benchmark. The code provided is used to generate the data that is loaded into the database and to generate the transactions and the data for the transactions. By providing code, the TPC hopes that it will be easier for more member companies to set up and run the benchmark. The TPC's intent is that test sponsors will spend time optimizing their products rather than coding and optimizing the driver for the benchmark.

Similarities to TPC-C

TPC-E is similar to TPC-C in these ways:

- The primary metrics for TPC-E are tpsE, $/tpsE and availability date. These metrics correspond to those of TPC-C: tpmC, $/tpmC, and availability date. Also, neither TPC-C nor TPC-E uses a scale factor, as TPC-H does.
- TPC-E and TPC-C both use an OLTP workload, although each is based on a different business model.
- Portions of the database scale in a linear fashion in both benchmarks.
- The transaction profile is held constant.

3 Overview of TPC-E

3.1 Brokerage Firm Model

As Figure 1 illustrates, customers generate transactions related to trade requests, account inquiries and market research. The brokerage firm sends trades to the market. The market returns the results of the trades and also sends a constant stream of the latest price for each security being traded. The brokerage firm returns the results of transactions to the customers.

Several models were considered for the workload. The brokerage firm model, which is based on input received from industry experts, such as Merrill-Lynch and Fidelity, was selected. This model met the volume/scaling and market relevance criteria for the benchmark. Industry analysts have mentioned that TPC-C, the current leading benchmark for database servers, is no longer as relevant to client OLTP workloads. One example of this lack of relevance is that for the current number one TPC-C result the database supported more customers than there are people on the planet. TPC-E needed to be able to have a meaningful number of customers in comparison to the performance of the database server being measured.

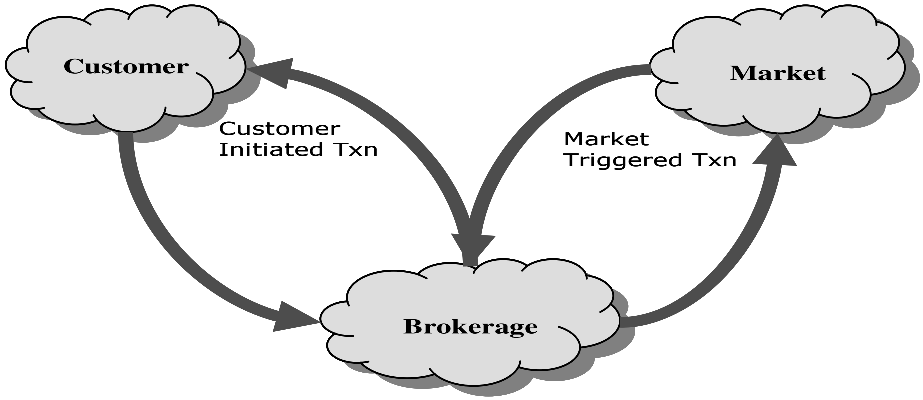

Fig. 1. TPC-E model – A Brokerage Firm

Another criticism leveled at TPC-C is that it has too few transaction types and that the transactions are too simple compared to the type of transactions clients do now. As processing power has increased, clients have added complexity to their transactions. The brokerage firm model is rich enough to enable the TPC-E benchmark to have more transaction types and more database tables than used in TPC-C.

There have been many technological advances since the TPC-C benchmark first came out roughly 17 years ago. The TPC-E workload is defined so that it can exercise some of these technological advances and features. The brokerage firm model lets us exercise features such as rich text from news articles about the companies whose securities are traded. Database Management System (DBMS) features (e.g., integrity constraints) that are commonly used today are also incorporated in the workload.

3.2 TPC-E Database Information

When loading the database for TPC-E, the benchmark sponsor chooses the number of customers based on the tpsE they are aiming for, keeping in mind that there are 500 customers per single tpsE. Customers can be loaded only in blocks of 1,000. Some of the other TPC-E tables scale based on the number of customers chosen.

The TPC provides code to generate the data for the TPC-E database. The TPC-E data generator uses names from a U.S. census and information from the New York Stock Exchange to generate people's names and company information. This makes TPC-E data look like normal data. TPC-C data concatenates "foo," "bar," and so on to generate names, resulting in "unnatural" looking names.

Trading in TPC-E is done by Accounts. Accounts belong to Customers. Customers are serviced by Brokers. Accounts trade Securities that are issued by Companies.

The total set of Securities that can be traded and the total set of Companies that issue Securities scales along with the number of Customers. For each unit of 1,000 Customers, there are 685 Securities and 500 Companies (with Companies issuing one to five Securities, mostly common shares, but some preferred as well).

All Companies belong to one of the 102 Industries. Each Industry belongs to one of the 12 market Sectors.

Each Account picks its average 10 Securities to trade from across the entire range of Securities. Securities to be traded can be identified by the security (ticker) symbol or by the company name and security issue.

Differences between Customer Tiers

The basic scaling unit of a TPC-E database is a set of 1,000 Customers. For each set of 1,000 Customers, 20% belong to Tier 1, 60% to Tier 2, and 20% to Tier 3. Tier 2 Customers trade twice as often as Tier 1 Customers. Tier 3 Customers trade three times as often as Tier 1 Customers. In general, customer trading is non-uniform by tier within each set of 1,000 Customers.

Tier 1 Customers have 1 to 4 Accounts (average 2.5). Tier 2 Customers have 2 to 8 Accounts (average 5.0). Tier 3 Customers have 5 to 10 Accounts (average 7.5). Overall, there is an average of five Accounts per Customer.

The minimum and maximum number of Securities traded by each Account varies by Customer Tier and by the number of Accounts for each Customer. The average number of Securities traded per Account is 10 (so the average number of Securities traded per Customer is 50). For each Account, the same set of Securities is traded for both the initial database population and for any benchmark run.

Customer Partitioning

TPC-E scales with Customers. It is conceivable that Customer information could be partitioned into groups of related Customers. This is called Customer Partitioning. The advantage of Customer Partitioning is that it increases locality of reference within each sub-group of Customers. Transactions relating to a particular set of Customers are directed to that set of Customers rather than to all Customers.

Trade Types

Trade requests are either Buy (50%) or Sell (50%). These are further broken down into Trade Types, depending on whether the request was a Market Order (60%) or a Limit Order (40%).

For Market Orders, the two trade types are Market-Buy (30%) and Market-Sell (30%). For Limit Orders, the three trade types are Limit-Buy (20%), Limit-Sell (10%) and Stop-Loss (10%).

Market-Buy and Market-Sell are trade requests to buy and sell immediately at the current market price, whatever price that may be. Limit-Buy is a request to buy only when the market price is at or below the specified limit price. Limit-Sell is a request to sell only when the market price is at or above the specified limit price. Stop-Loss is a request to sell only when (or if) the market price drops to or below the specified limit price.

If the specified limit price has not been reached when the Limit Order is requested, it is considered an Out-of-the-Money request and remains "Pending" until the specified limit price is reached. Reaching the limit price is guaranteed to occur within 15 minutes based on benchmark implementation details. The act of noticing that a "Pending" limit request has reached or exceeded its specified limit price and submitting it to the market exchange to be traded is known as “triggering" of the pending limit order.

Effects of Trading on Holdings

For a given account and security, holdings will be either all long (positive quantities) or all short (negative quantities).

Long positions represent shares of the security that were bought (purchased and paid for) by the customer for the account. The customer owns the shares of the security and may sell them at a later time (hopefully, for a higher price).

Short positions represent shares of the security that were borrowed from the broker (or Brokerage) and were sold by the customer for the account. In the short-sale case, the customer has received the funds from that sell, but still has to cover the sell by later purchasing an equal number of shares (hopefully at a lower price) from the market and returning those shares to the broker.

Before the database is loaded, there are no trades and no positions in any security for any account. The TPC provides code to generate the data for the database. This data generation code simulates running the benchmark for 300 business days of initial trading, so that the initial database will be ready for benchmark execution. The data-generation code also generates data for daily market closing price information for five years of five-day work weeks, and five years’ worth of quarterly report data for all the companies.

If the first trade for a security in an account is a buy, a long position will be established (positive quantity in HOLDING row). Subsequent buys in the same account for the same security will add holding rows with positive quantities. Subsequent sells will reduce holding quantities or delete holding rows to satisfy the sell trade. All holdings may be eliminated, in which case the position becomes empty. If the sell quantity still is not satisfied, the position changes from long to short (see above).

If the first trade for a security in an account is a sell, a short position will be established (negative quantity in HOLDING row). Subsequent sells in the same account for the same security will add holding rows with negative quantities. Subsequent buys will reduce holding quantities (toward zero) or delete holding rows to satisfy the buy trade. All holdings may be eliminated, in which case the position becomes empty. If the buy quantity still is not satisfied, the position changes from short to long.

Database Tables

TPC-C had only nine database tables. Most client databases have more than nine tables. TPC-E has many more tables than TPC-C. The TPC-E database tables can be grouped into four categories:

- Customer – tables containing customer-related information
- Broker – tables containing data related to the brokerage firm and brokers
- Market – tables containing data related to the exchanges, companies, and securities that create the financial market
- Dimension – tables containing generic information that is referenced by multiple fact tables

Table 6. Customer tables

Customer Tables	
ACCOUNT_PERMISSION	Who can execute trades for accounts
CUSTOMER	Customer information
CUSTOMER_ACCOUNT	Accounts for each customer
CUSTOMER_TAXRATE	Tax rates each customer pays
HOLDING	Customer account's security holdings
HOLDING_HISTORY	History of how trades changed holding positions
HOLDING_SUMMARY	Aggregate of customer account's security holdings
WATCH_ITEM	List of securities customers are tracking on their watch lists
WATCH_LIST	Customer's security watch lists

Table 7. Broker tables

Broker Tables	
BROKER	Broker information
CASH_TRANSACTION	Cash transaction information
CHARGE	Information about trade charges
COMMISSION_RATE	Commission rate information
SETTLEMENT	Trade settlement information
TRADE	Trade information
TRADE_HISTORY	History of each trade through various stages
TRADE_REQUEST	Pending limit trades
TRADE_TYPE	Valid trade types

During the benchmark run:

- All customer tables are read.
- CUSTOMER_ACCOUNT balance is updated.
- HOLDING_SUMMARY table is updated.
- Records are appended to the HOLDING_HISTORY table.
- Records are updated, deleted from and inserted in the HOLDING table.

During the benchmark run:

- All broker tables are read.
- BROKER table is updated with the number of trades a broker has executed and the commission the broker has earned so far.
- Records are appended to CASH_TRANSACTION, SETTLEMENT and TRADE_HISTORY tables.
- Records are appended and updated in the TRADE table.
- Records are inserted in and deleted from the TRADE_REQUEST table.

During the benchmark run:

- All market tables are read.
- The LAST_TRADE table is updated many times a second.

Table 8. Market tables

Market Tables	
COMPANY	Information about companies with publicly traded securities
COMPANY_COMPETITOR	Information for the competitors of a given company and the industry in which the company competes
DAILY_MARKET	Daily market statistics for each security
EXCHANGE	Financial exchange information
FINANCIAL	Information about a company's quarterly financial reports
INDUSTRY	Industry information
LAST_TRADE	Latest price and trading volume for each security
NEWS_ITEM	News items of interest
NEWS_XREF	Cross-reference of the news items to companies mentioned in the news item
SECTOR	Market sector information
SECURITY	Information about each security traded on any of the exchanges

During the benchmark run, all the dimension tables are read.

Table 9. Dimension tables

Dimension Tables	
ADDRESS	Address information
STATUS_TYPE	Status values
TAXRATE	Tax rate information
ZIP_CODE	Zip code information

3.3 TPC-E Transactions

TPC-C had only five transactions: New-Order (45% of the transaction mix), Payment (43%), Delivery (4%), Stock-Level (4%), and Order-Status (4%). The tpmC metric equaled the number of New-Order transactions done per minute. Because almost half of the transaction mix was New-Orders, configuration optimizations could change the metric significantly.

TPC-E has 10 transactions that are part of the maintained transaction mix, and two other transactions. Trade-Result is the transaction that is counted for the tpsE metric. Trade-Result is only 10% of the transaction mix. The Data-Maintenance transaction, which runs once per minute, is not part of the maintained transaction mix. The Trade-Cleanup transaction is only run once before starting a benchmark run. The following sections provide a short description of each transaction.

Broker-Volume (4.9% of the Transaction Mix)
The Broker-Volume transaction is designed to emulate a brokerage house's "up-to-the-minute" internal business processing. An example of a Broker-Volume transaction would be a manager generating a report on the current performance potential of various brokers. The transaction is a business intelligence type of query that only does reads and is CPU-heavy.

Customer-Position (13%)
The Customer-Position transaction is designed to emulate the process of retrieving the customer's profile and summarizing their overall standing based on current market values for all assets. This is representative of the work performed when a customer asks the question "What am I worth today?" The transaction is a read-only transaction.

Market-Feed (1%)
The Market-Feed transaction is designed to emulate the process of tracking the current market activity. This is representative of the brokerage house processing the "ticker-tape" from the market exchange. The transaction is a read/write transaction.

Market-Watch (18%)
The Market-Watch transaction is designed to emulate the process of monitoring the overall performance of the market by allowing a customer to track the current daily trend (up or down) of a collection of securities. The collection of securities being monitored may be based upon a customer's current holdings, a customer's watch list of prospective securities, or a particular industry. The transaction is a read-only transaction.

Security-Detail (14%)
The Security-Detail transaction is designed to emulate the process of accessing detailed information on a particular security. This is representative of a customer doing research on a security prior to making a decision about whether to execute a trade. The transaction is a read-only transaction.

Trade-Lookup (8%)
The Trade-Lookup transaction is designed to emulate information retrieval by either a customer or a broker to satisfy their questions regarding a set of trades. The various sets of trades are chosen such that the work is representative of:

- Performing general market analysis
- Reviewing trades for a period of time prior to the most recent account statement
- Analyzing past performance of a particular security
- Analyzing the history of a particular customer holding

The transaction is a read-only transaction. This transaction generates a lot of disk IO because it looks for older records that don't tend to be in memory because they were not used recently.

Trade-Order (10.1%)
The Trade-Order transaction is designed to emulate the process of buying or selling a security by a Customer, Broker, or authorized third-party. If the person executing the trade order is not the account owner, the transaction will verify that the person has the appropriate authorization to perform the trade order. The transaction allows the person trading to execute buys at the current market price, sells at the current market

price, or limit buys and sells at a requested price. The transaction also provides an estimate of the financial impact of the proposed trade by providing profit/loss data, tax implications, and anticipated commission fees. This allows the trader to evaluate the desirability of the proposed security trade before either submitting or canceling the trade. The transaction is a read/write transaction.

Trade-Result (10%)
The Trade-Result transaction is designed to emulate the process of completing a stock market trade. This is representative of a brokerage house receiving from the market exchange the final confirmation and price for the trade. The customer's holdings are updated to reflect that the trade has completed. Estimates generated when the trade was ordered for the broker commission and other similar quantities are replaced with the actual numbers, and historical information about the trade is recorded for later reference. The transaction is a read/write transaction and is counted as the tpsE metric.

Trade-Status (19%)
The Trade-Status transaction is designed to emulate the process of providing an update on the status of a particular set of trades. It is representative of a customer reviewing a summary of the recent trading activity for one of their accounts. The transaction is a read-only transaction.

Trade-Update (2%)
The Trade-Update transaction is designed to emulate the process of making minor corrections or updates to a set of trades. This is analogous to a customer or broker reviewing a set of trades, and discovering that some minor editorial corrections are required. The various sets of trades are chosen such that the work is representative of reviewing:

- General market trends
- Trades for a period of time prior to the most recent account statement
- Past performance of a particular security

The transaction is a read/write transaction. This transaction generates a lot of disk I/O because it looks for older records that don't tend to be in memory because they were not used recently.

Data-Maintenance (Runs Once per Minute)
The Data-Maintenance transaction is designed to emulate the periodic modifications to data that is mainly static and used for reference. This is analogous to updating a customer's e-mail address or other data that seldom changes. The transaction is a read/write transaction.

Trade-Cleanup (Runs Only One Time Before the Benchmark is Started)
The Trade-Cleanup transaction is used to cancel any pending or submitted trades from the database. The transaction is a read/write transaction.

4 Transition to TPC-E

4.1 Current Landscape

The momentum for TPC-E is increasing. To date, 26 TPC-E results have been published from six hardware vendors on systems ranging from 1 to 16 processors. During the year 2008, 17 TPC-C results were published from 7 different hardware vendors compared to 14 TPC-E results from 6 vendors- fairly even. So far in 2009, there are 4 TPC-C publishes from 2 vendors compared to 6 TPC-E publishes from 5 vendors. And TPC-E so far only has publishes on Microsoft Windows and SQL Server. Once the other database vendors get up to speed with TPC-E, the publications should really take off.

4.2 Next Steps

As expected, TPC-C and TPC-E results are coexisting. With the clear benefits of TPC-E, especially the much lower cost of benchmarking in this economy, it has good traction and is expected to supplant TPC-C. However, before that can happen, some non-SQL Server TPC-E publications are needed. This will take time. Even with the TPC-supplied benchmarking code, it takes quite some time to develop and test a benchmark kit and then use it to tune the hardware and software to get a good result. But this is exactly why TPC-E was developed. The hardware and software vendors are using it to help optimize their products for uses that more closely mimic those of typical clients. It may be a while, but the results of this work will be seen. Once results are published using other database products, expect to see back and forth result leadership as the hardware and software become more tuned for the benchmark.

In the meantime, it may be tempting to try to compare TPC-C and TPC-E results. However, TPC-C and TPC-E results are not directly comparable. They are different workloads, and they each scale slightly differently. For instance, with TPC-C, greater performance gains are realized when memory is doubled—not so with TPC-E. But TPC-E does realize greater performance gains from increasing processor frequency or adding processors than does TPC-C. There is no magic formula to translate a database server's TPC-E score to a TPC-C score or vice versa.

References and Additional Information

1. TPC-E Specification, `http://www.tpc.org`
2. TPC-PR TPC-E presentation, available from members-only side, `http://www.tpc.org`