Transaction Processing Performance Council (TPC)

OLTP Benchmark Proposal Development Guidelines (PDG)

Revision 1.0 – December 4, 2001

Introduction

The Transaction Processing Performance Council (TPC), an Industry Standards body dedicated to the development and dissemination of database workloads, is requesting proposals for the next On-Line Transaction Processing (OLTP) workload to be developed into a new TPC Benchmark. The TPC has over a decade's worth of experience in providing industry standard workloads like TPC-A/B (simple OLTP), TPC-C (complex OLTP), TPC-D (decision support), TPC-H (Ad-hoc decision support), TPC-R (business reporting) and TPC-W (web based e-commerce).

This document addresses the requirements for an OLTP workload. The requirements cover all aspects of a proposed benchmark, including; benchmark overview and design goals, database design and population, transaction profiles, run-time and statistics reporting, configuration limits and reporting, auditing and full disclosure reports. The presentation of a response should be visual (presentation) along with a draft benchmark specification (similar to the above mentioned benchmarks).

Brief TPC History

On-Line Transaction Processing systems/environments have been used commercially for decades. It was not until the late Eighties the TPC formed to develop a standard by which to measure OLTP performance. The first workload was TPC-A (now obsolete) and was a complete success. TPC-C was proposed to the TPC by one of its member companies, accepted as a draft and developed into what is known throughout the industry as the premiere OLTP industry standard workload. However, as TPC-C was gaining in acceptance and recognition as a standard workload, the typical customer environment for OLTP work exceeded the components tested by TPC-C.

While OLTP workloads have evolved and new technologies have entered into mainstream usage by companies, OLTP remains the cornerstone by which most businesses do business. Whether the business is an e-commerce store or a grocery store, the heart and soul of its environment is OLTP. The TPC is looking to develop a new workload that better reflects the changes to OLTP processing over the past decade. The TPC is soliciting responses to this Requirements Document as outlined in the following pages.

Overview of Requirements

A successful response should incorporate standard OLTP features like a commercial database manager, a complex database schema, a rich transaction set, and a complex, yet easy to comprehend business model. Along with these traditional OLTP requirements, the PDG defines requirements in the following areas: security elements, availability concepts, manageability features, a relevant simple metric, an extensible database and transaction environment, backup/restore requirements, non-uniform database population and access and user input loading while maintaining a simple user context.

Benchmark Requirements

This PDG is soliciting benchmark proposals for a new OLTP workload. The purpose of any benchmark workload should be to:

- Provide a repeatable environment for comparison of results
- Stress components of the environment to identify areas for improvement
- Provide a vehicle for marketing to disclose the performance of an environment and compare/contrast to competitors.

This PDG is soliciting benchmark proposals to expand beyond the normal primary metrics of the TPC, price and price/performance. Secondary metrics are encouraged for measurement and compliance with some of the requirements. While the DBMS is the core component being exercised by OLTP systems, it is no longer the only component of the environment that affects performance.

The benchmark proposal in response to this PDG must be an OTLP database centric workload. The expectation is for the proposal to have a high volume of transactions, both simple and complex transaction types, and a complex transaction and database schema. Representative response time limits must be imposed. The proposal should strive to balance the network and disk I/O requirements with the computing requirements. The PDG details more requirements than just the transaction details. The additional requirements address non-database needs of OLTP environments. The specifics for each requirement are spelled out in detail below. While it may seem highly complex on the surface, the objective is that each requirement is but a part of the overall environment, one that complements the OLTP environment instead of dominating or distracting from the purpose of the benchmark.

The PDG is divided into three main categories with a total of 15 major requirements. The main categories are: Generic TPC Benchmark Characteristics, Workload Characteristics, and Operational Environment Characteristics. The TPC characteristics are found in every TPC benchmark and create the cornerstone of the TPC’s workloads. Workload characteristics are the individual distinguishing properties that allow workloads to target certain technologies and customer sets. The newest sets of characteristics to be incorporated into TPC workloads are the Operational Environment requirements. These are being developed to incorporate emerging trends and technologies in customer environments.

Generic TPC Benchmark Characteristics

These requirements form the foundation upon which all TPC benchmarks are built. The TPC has developed several widely accepted and well-known workloads, but the workloads themselves do not guarantee acceptance or generate trustworthy results. It is a combination of the workloads with the following requirements that make the TPC benchmarks successful. Market Relevance is the basic representation for acceptability of a workload. Providing easy to relate to metrics like transactions per minute (tpmC) helps with understanding. TPC workloads are used for greater purposes than just marketing material; they provide excellent repeatable measurements to track improvements and distribution of the technologies to the customers. The audit requirements for each TPC workload are a representation of the minimum set of features necessary to ensure the reliability of the environment and its data. The TPC has proven the success of its benchmarks over the years using these requirements to build stable, repeatable and successful workloads.

Market Relevance

Performance benchmarks have provided valuable product performance data to our customer community. They also encourage the technical community to continue to improve products to better serve the market place. Performance benchmarks come from various sources, from public consortiums such as TPC and from private vendors such as SAP. There are multiple aspects of performance. Each performance benchmark measures different aspects of the system performance, addressing a different market segment. For example, TPC-H measures the decision support performance of database products. TPC-W focuses on the end-to-end e-business system performance. SAP benchmarks measure the various performance aspects of the ERP system. Of course, TPC-C measures the database performance in an OLTP environment.

Performance benchmarks are not static; they need to continue to evolve to keep up with advancing technologies and customer demands. TPC-C is considered the standard OLTP benchmark by the market place. The OLTP environment has undergone many changes since the introduction of TPC-C and we feel there is a need to create a new benchmark to reflect the changes in the market place. This brings us to the purpose of this PDG. We are looking for an OLTP benchmark to address the new business realities in today's OLTP environment.

The business models and workloads for all TPC Benchmarks are well understood and are executed under vigorous rules. It is easy to explain to a non-technical person what each benchmark models and what it measures. Customers can extrapolate these benchmark results to help predict the performance of their own systems. In an international market place, ideas and concepts that transcend borders and languages are keys to success. The proposed benchmark must be database centric benchmark. It should have an easy to understand business model. The benchmark must provide clear and useful performance reference points to customers. Although the benchmark may not mirror any specific customer workload, customers should be able to use benchmark results to gain insights into how the system will perform in their environments.

The proposed OLTP benchmark should be scalable across a wide range of systems (e.g. entry-level servers, mid-range servers, mainframes, clusters). All systems run the same workload and generate results that are comparable. This does not mean the complexity of the workload stays constant. The workload complexities are hidden in the System Under Test (SUT) while the profile from the user's perspective remains the same. The results are comparable regardless of the complexities of the different hardware and software configurations. With results from many different systems, customers can find a wealth of relevant comparisons.

The cost of ownership is important to customers; although it is unlikely the customer will have an exact replica of the benchmark system, the price/performance offers relative comparison for the customer. The benchmark results should enable customer and marketing audiences to make fair performance and value (price /performance) comparisons.

The proposed benchmark should measure peak OLTP performance. It should also measure the overheads of other key challenges to maintain OLTP systems. Measuring peak OLTP system performance is just as important as before, however, customers must also address areas such as system availability and security. Incorporating these functions makes the benchmark more relevant to customers facing similar challenges.

In summary, with respect to market relevance, the proposed OLTP benchmark must satisfy these requirements:

- Have a simple business model which is easily understood by non-technical audiences such as marketing and user groups
- Provide a comparable OLTP performance metric.
- Feature a primary performance metric measuring an OLTP workload
- Provide Price and price/performance of SUT

In addition to the requirements above, the benchmark proposal should take into consideration the following additional features:

- Address the challenging issues of OLTP systems facing our customers.

Simple Metrics

A Metric is a standard of measurement. In this case we need to set a standard of measuring the performance of the System Under Test (SUT). The metric must be simple enough for the market to understand and relate to, but should represent a fairly complex workload being performed by the SUT. A performance metric shall be a basis for comparing, evaluating and assessing the quality of the SUT. Great care, should therefore, be taken to ensure that the metric is easy to measure, to understand, and is comparable.

The proposed benchmark must identify a single primary metric, which should depend on the benchmark model selected. A secondary metric that measures the roundtrip response time across the SUT must also be incorporated. The current TPC-C benchmark has successfully used Throughput and Price per Throughput as a primary metric.

The current TPC-C metric is simple and well understood by the industry but the workload it measures does not fully represent advances in OLTP. The *Transaction Complexity* section of the PDG covers the complexity of the workload but responders must address the aspects of the workload that are measured. The transactions that best represent the model should be weighted and measured in an environment where the transactions should have maximum access to resources. That measurement will constitute the primary metric. The idea is to capture the performance of the system when it is devoted to its core transactions.

An additional measurement that shows the performance of the system with common housekeeping tasks (like backups) is encouraged and should be reported in the Full Disclosure Report (FDR). The measurement should not be part of the primary metric.

The behavior of the system when it loses some of its resources or when it is under considerable strain is another area that must be considered; see the *Availability Section*. In summary, the PDG must define the following:

- Primary metric – Throughput, Price/Throughput
- Secondary metric – Roundtrip Response time across the SUT, Performance loss

Additionally, it would be nice to report

- DB full backup time and cost
- Cost of system
- IO stats
- NW stats
- CPU stats
- Loss of connections
- DB size
- The cost and time of performing a full backup

A performance metric defines the applicability and suitability of a benchmark. Considerable attention will therefore be given to proposals that feature a simple metric that is comparable across a diverse range of environments but captures the complexity of a modern day OLTP system.

Technology Driver

It is not sufficient for a benchmark to simply measure the current state of technology. OLTP technology will move forward, with or without the benchmark's influence, but as TPC-C has demonstrated, the benchmark itself can be a strong driver of technology. The race itself inspires the investment in R&D necessary for advancing the technology used in the benchmark.

On the other hand, the benchmark should not impede technological progress, by too rigidly specifying requirements on the parts used to make up the implementation. The

TPC approach has always been to specify a problem to be solved, and encourage, but not require certain specific means to solve that problem.

This benchmark is an OLTP benchmark. The technology that it should drive forward should relate to OLTP-class problems.

Most OLTP systems do not simply serve users sitting at terminals, as they did 30 years ago. OLTP systems serve a variety of interlinked information systems, from direct Web users to inter-related business applications. Many OLTP applications have to be able to accept simultaneous requests from a number of data sources, in different formats, and use the same application code to satisfy them. Standard protocols for inter-system OLTP traffic are being defined, and new OLTP applications will have to process these formats efficiently.

One area of OLTP that has been overlooked in the past is queuing. Although TPC-C defines the deferred Delivery transaction in terms of queuing, there is no requirement to use a commercial, durable queuing product. This benchmark should raise the bar and drive performance improvements in this important area of OLTP software. OLTP systems also now use advanced technologies, such as two-phase commit, when more than one application has to be involved in an atomic transaction.

OLTP systems have more stringent availability requirements than they did years ago, as businesses are expected to be available 24x7. Reliable storage systems are a requirement for the new benchmark.

Referential integrity should be maintained by the database, rather than expecting the application code to code RI as business rules. Security is extremely important in this world of interconnected applications, and Internet users. The benchmark should encourage the use of secure communications and database encryption, so that these technologies can be improved in the marketplace.

The response to this PDG must include the following characteristics:

- Multiple access modes, e.g., Browser, Wireless phone and SOAP, all executing common business logic.
- Reliable Queuing, for deferred processing or workflow.
- A standard industry protocol for inter-business OLTP (e.g., SOAP/XML)
- Referential Integrity exercised in the DBMS
- RAID storage, or the equivalent
- Rapid recovery from failures

Further consideration will be given to proposals that include the following optional characteristics:

- Interaction requirements between two or more databases that suggest the use of 2-phase commit technologies.
- Incorporation of a business requirement that suggests the use of a managed run-time environment (e.g. Java, C#, etc) for at least one application component.

- Database encryption

These requirements make up a modern, reliable business OLTP system, that works in harmony with other internal and external applications, and is mission critical, i.e., must be available, and must recover from diminished service rapidly.

The offered proposal should go well beyond TPC-C in the above areas, so that vendors are encouraged to further improve the technologies that implement such capabilities.

Audit Requirements

Overview

An independent (*) audit of the benchmark results by an auditor certified (**) by the TPC is required. An audit checklist has to be provided as part of any proposal. The purpose of the audit is to:

- Ensure compliance with the specification.
- Verify that the benchmark has passed the minimum requirements.
- Ensure the spirit and wording of the benchmark specification are met.
- Make sure that the result can be repeated externally.

It is not the purpose of the audit to check for any obvious system functions (e.g.: does a write work), but it is up to the auditor to verify whatever they find important.

(*) The term "independent" is defined as: "the outcome of the benchmark carries no financial benefit to the auditing agency other than fees earned directly related to the audit." In addition, the auditing agency cannot have supplied any performance consulting under contract for the benchmark under audit.
(**) The term "certified" is defined as: "the TPC has reviewed the qualification of the auditor and certified that the auditor is capable of verifying compliance of the benchmark result."

Requirements regarding the audit process

A response to this PDG must contain a minimum audit checklist, which is a list of minimum checks an auditor has to verify for each benchmark. See Chapter 9 of the TPC-C spec for a sample checklist. The proposal must make it possible to check all major components of the benchmark.

The audit checklist must include (but definitely not be limited to) the following requirements:

- The auditor has to make sure that the tested product is the same as benchmarked product and that the software that is used is available within a certain timeframe.
- The metric (all required) for the benchmark has to be verified and the validity of the method used to measure the response time at the Driver has to be verified.
- Tests similar to the ACID tests as defined in Clause 3 of the TPC-C benchmark.
- Detailed audit tests for Security, Manageability, Availability, Referential Integrity and other major portions of the benchmark.

- Additional tests for new components of the benchmark.
- It must be possible to verify that the database has been built according to the specification and that the database is initially populated with the properly scaled required population.
- It also must be possible to verify the correct cardinalities of all the database tables, at the start of the benchmark run as well as at the end of it, and that the growth in each growing table, in particular, is consistent with the number and type of executed transactions.

Optional requirements for the proposed specification are:

- The auditor checklist for each benchmark might be made available in the FDR (Audit should be more fail-safe).
- In order to verify the audit requirements - the specification may require trace points and logging points in the benchmark code implemented which can be enabled with external parameters.
- Additional management software may be required, that enables additional SW accounting (like what disks are connected to the system, how many processors are connected, network configuration).

Workload Characteristics

The characteristics described in this section are what differentiate this benchmark from other TPC Benchmarks. These requirements determine what is being tested, how relevant the workload is, how long it will last as a standard and how technology changes will interact these requirements.

Scalability

Scalability is a difficult matter to represent in an artificial workload. For the purpose of this PDG, scalability is defined as *"How well a solution to some problem will work when the size of the problem changes"*. It is quite common to find that a solution to a problem at one size is under scaled or inappropriate as the problem grows in size. The longevity of a benchmark as well as how representative it is and how comparable results are over time is greatly affected by the workload's scalability. A successful workload will generate meaningful results at both the low-end and high-end of the computing spectrum. This means that the work generated is representative of both the environment and complexity of a wide range of computing solutions. Therefore it is critical for a proposal to this PDG take great care in planning the lifespan, comparability, database and workload growth, technology growth, and performance curve.

The current TPC-C benchmark incorporates a linear scaling model; for each additional warehouse added to the database a fixed number of rows are added to most of the remaining tables. Also, for each added warehouse 10 emulated users are added. Thus the performance "scaling" is almost linear because the unit of work per transaction remains constant. (The term "unit of work per transaction" as used in this PDG and in TPC specifications refers to the logical steps of a transaction as performed by the application or the database engine.) This leads to greater longevity and comparability of results over a long period of time, but also impacts the benchmark's representation of OLTP environments.

A response to this PDG must incorporate the following components directly concerning scalability into the proposed workload:

- Linear growth of the database, or portions thereof, with the performance growth of the system
- A constant, well defined transaction profile.
  - Any change in the amount of work per transaction is a by-product of the scaling of the database size and/or transaction rates
- Representative user scaling
  - While user population may scale with database size, it is more important that the representation of the user population scale in a realistic fashion
- Typical transaction environment
  - The ability to generate an effective transaction and database load on a wide range and size of computers; e.g. from single systems to clusters.
- All results published must be comparable

Responders may wish to scale at least one transaction's database to the performance of the system.

Linear growth of the database is defined as: as the performance of the computing environment increases, the number of database records in the database and used during the test increases accordingly. This is done to maintain a one to one relationship of performance growth with database growth. Please see the *Database Complexity* section for more detailed requirements on the database structure and environment.

The next requirement is for a constant, well-defined transaction set. TPC-C today provides an excellent example of a well-defined transaction profile set with constant load characteristics. This PDG is encouraging transactions that have been well thought out, have a good correlation or relationship with the work being done by an OLTP system, and whose transaction profile remains constant over the lifespan of the workload. This does not preclude transactions whose working set or instruction set increases with system performance, just that the transaction profile or logical unit of work does not change with scaling of performance or system sizes. Please see the *Transaction Complexity* section for more details on this requirement.

The number of users does not always scale perfectly with increased performance. It is reasonable to expect environments where additional services or features are added in a transaction profile, instead of just doubling the number of users. This is a difficult subject to capture in a synthetic workload, but one that should be given serious consideration. It is not reasonable to assume that the number of users an environment can support will scale according to Moore's Law, but rather the amount of work being performed should scale. In TPC-C there exist a 1:1 relationship between the amount of work being performed and the number of users as the tpmC increases, so does the number of emulated users. This is not always the case in OLTP production systems. Many environments today are increasing the amount of work performed by the computing environment while maintaining roughly the same performance. This is accomplished by incorporating new features or workload requirements into traditional OLTP functions and by making more traditional decision support queries operate in the OLTP environment. Additional details and requirements are available in the *Transaction Complexity* and *User Context* sections.

One of the strengths of TPC workloads is the representative nature of them. This means that the work being done by the environment is believable and representative. Any proposal in response to this PDG must have a transaction environment that is representative of the size system the work would be performed upon. In other words, the workload should be realistic and representative of the computing requirements anywhere from a single smaller system all the way to a large multi-system clustered environment or mainframes. The workload should stress the components of the environment no matter the number of cpus.

Comparability is one of the TPC's major objectives in benchmark design. A benchmark that is comparable across the spectrum of results leads to greater marketing opportunities and acceptance of a workload. Benchmarks that are marketed fairly and where all results are comparable and understandable will be better accepted and have greater longevity. TPC-C is an example of a highly comparable, well-accepted workload by the industry.

A comparable workload is one whose metrics, performance, and workload are similar. Horsepower is a measure of force over time (550 ft lbs per second). By applying this measure to an engine, manufacturers are able to determine the output of different types and designs of engines. The same approach to computers is made with the TPC benchmarks. By applying a constant measurement to systems, a transaction rate over time is measured, reported and can be compared to other systems.

One element of this that seems to be inconsistent with comparability is scaling the workload to the performance of a system. Consideration will be given to proposals that incorporate a transaction(s) that perform more work as the performance of a system increase, while maintaining a constant transaction profile. An example of this concept; The TPC-C Stock-Level transaction executes against the last 20 orders submitted for a warehouse. To make this transaction scale while maintaining it profile, the Stock-Level transaction could operate against the last order for every warehouse configured. As the number of warehouses increases on a system, the number of orders used by the Stock-Level transaction increases and thus the amount of the system's resources increases.

Scalability is a very important issue in this PDG and will be referred to in more sections. Attention must be paid to the balance of transactions, the balance of the database growth, and to the comparability and longevity of the results.

Extensibility

Extensibility as used in this PDG refers to a workload's ability to grow and change to mirror changes in the environment it is designed to emulate. However, this can also lead to problems with comparability of results.

The TPC is looking for a workload with:

- A well-defined transaction set
- A stable definition of the primary metric that does not change over time

Optional requirement to be considered:

- A roadmap to extend the workload over a five to ten year period

The roadmap can be as simple as changing the mix of transactions, the intensity of transactions (by changing compute intensive or I/O intensive code) or by adding additional features, requirements or transactions. Once a workload has become established it is nearly impossible to modify it without the changes being planned as part of the development. The areas of a new workload to concentrate on for future extension are the database schemas, the individual transaction complexities (adding compute intensive requirements), and adding new features or technologies and testing of the features or technologies to the workload. The core transaction set must not change over the lifetime of the workload. The roadmap will help to provide guidance to the TPC as to how the workload could be extended to incorporate new technologies, provide new verification and functionality testing of new products and possibly extend the workload with new transactions. These extensions are not primary to the proposal, but could enhance it with guidance for the TPC on how to evolve the workload over time.

User Context

The term *User Context* in this benchmark refers to a software layer that simulates or emulates the part of the database application that collects real-world events and initiates data processing as a result. The *User Context* provides a response indicating the processing completed successfully or an error occurred.

The intent of requiring the user context is to exercise the libraries that would normally be used in compiling and linking a real-world OLTP application. The actual user interface, e.g. HTML forms, Terminal I/O, bar code reader, etc, will not actually be present. An acceptable replacement for this actual user interface could be a network packet with defined fields that the user application would accept as input and transmit as a resultant output. The generation and reception of these I/O packets must be from a system that is capable of generating the required workload and measuring the response-times of the returned results. This traditionally has been accomplished with separate driving systems commonly called Remote Terminal Emulators (RTE) or Remote Browser Emulators (RBE). Previously these systems would actually generate user input, and accept user readable output. User readability is not required and data from these driving systems must be encoded and decoded in a predefined format.

The driving systems need to be able to measure the simulated response times and be able to generate reports to meet the audit requirements and reporting. The program used to drive the workload will also require auditing to certify that load generation and reporting is compliant with the specification.

The specification must specify how the user context is simulated and the means by which the load is generated and measured. The following is required in the specification:

- Generate workload via a remote system that simulates a user or other real-world event. The actual user interface must not be present.
- Synchronously measure response times of the business transaction processing accomplished on the system under test.
- Require the application code to utilize the libraries normally used to access or connect to the data processing software (DBMS).
- Have a defined format for all the messages (packets) sent and received by the application layer software (user application) and the driving system(s).
- Provide requirements for auditing and reporting of the driver systems.
- Require full-disclosure reporting of the simulated application source-code.

Transaction Complexity

Transaction Complexity refers to the level of intricacy in the "typical" customer's workload. Everyone recognizes that different customers run different applications, and that they are all unique in at least some way. There is a common set of application features that are found in many, if not most, OLTP applications. A comprehensive benchmark captures and exercises as much of this set as possible.

Not all aspects of a customer's applications or all typical customer application aspects can be captured in a benchmark. The objective is to make the benchmark application complex enough to capture interesting aspects of OLTP. This makes the benchmark representative, realistic and relevant to customer needs.

The TPC-C benchmark has five transaction types that capture a number of aspects relevant to OLTP applications (e.g. online order entry, inventory management, debit/credit). Those aspects must be included in the new benchmark. In addition, we want to expand the scope to new aspects including additional resource intensive queries and queries with depth (i.e. "parts explosion" queries). This requires the addition of several new transaction types. At least seven or more transaction types must be defined, although that is by no means a limit. If a feasible method can be devised to generate considerably more queries relevant to a business problem, that solution would be considered.

Databases ease access to records by using redundant, index data structures. Sometimes a single path is sufficient to optimize access. But often the same sets of records are accessed in multiple ways. This leads to the use of secondary indices. Of course, the use of any indices is an implementation choice, and should not be directly specified. But the specification should frame a business problem that will naturally lead to use of secondary indices.

The TPC-C transactions are few (5), and focused strictly on core OLTP. A more realistic workload would contain some real-time analysis reporting queries in the overall mix. These would be more complex or "heavy" transactions, not unlike those found in decision support (DS) applications. The objective is not to create a DS type workload, but instead to reflect the growing sophistication of OLTP transactions (e.g., a credit card approval transaction may perform a DS like query to verify that the transaction is consistent with the users buying habits).

The database structures best supported in relational databases are very flat, i.e., they have a low level of hierarchical dependencies. In particular, "parts explosion" problems (also known as "bill of materials" problems), where a part is made up of a number of components, which are also made of components, and so on, can be problematic for relational databases. TPC-C, in particular, has a flat dependency structure, and does not stress this aspect of the system. Object-oriented databases claim to deal with such object connections more efficiently.

The proposed specification should address the real-world problem of managing parts-explosion-like database structure. We want to be able to measure behavior not simply for different numbers of parts, but also for different levels of connections and fan-outs. In particular, we need to understand the effect on performance of creating, selecting, retrieving, and modifying such "exploded" objects.

While the benchmark must stand on its own, we cannot ignore future extensibility. We would prefer the core transactions be constructed so that it is easy to see how they might be added to or extended in future benchmarks.

In summary, we require the specification to:

- Include common OLTP application features found in TPC-C
- Expand upon these key aspects to make the workload more realistic by:
    - Addition of at least 2 new transaction types, some "heavy"
    - Use of Secondary Indices
- Define a core transaction set which could be added to in time

In addition, although we do not require it, we would like to:

- Use "parts explosion" database elements
- Use some real-time analysis reporting queries

Load Variances

This section on Load Variances will detail the requirements for any proposal to have regarding initial database population data skew and the arrival rates and data locality of the workload. Representing the inherent variability found in data skew, arrival rates and data locality is difficult but very relevant for any workload. The closer a workload can emulate the activities of an environment, the more reliable the workload and environment become. The minimum requirements for Load Variances are:

- Coarse grain changes in data locality (hot spots) and data skew
- Medium grain changes for arrival rate and user load
- Medium grain changes for transaction mix variation

Any examination of a phone book or listing by family name throughout the world will reveal an uneven distribution of names. This is an example of data skew. There will be far more families of one name in a particular region than in others. For example, the numbers of “Jones’” are much greater in Britain than are “Lopez”. But in Mexico, “Jones” will be an uncommon family name. This non-uniform distribution is quite common in many areas besides family name. The proposal should have several areas of the database tables affected by data skew during the database build. The input data to the transactions should be affected by data skew or data locality. Data “hot spots” are created when consumers are attracted to products during cycles. An example is back to school clothes, backpacks, notebooks, etc. Non-uniform run-time data entry distribution requirements should be representative of the business model. TPC-C provides an example and well-tested implementation of data skew both in the database build and run-time environments.

An important run-time consideration is the ability to “dial” the rate of transactions and their “mix”. “Dialing” simply means the capability of changing the granularity of a component without requiring changes to the underlying applications. The TPC is looking for a workload with the capability to adjust the arrival rate and mix with “medium” grain changes. This basically means that an adjustment will have an immediate and noticeable change.

The benchmark must define a model that provides for a repeatable steady-state measurement interval. The arrival rate of transactions is defined as the number of new transactions initiated within a given time period. Non-uniform distribution of arrival rates can be used to introduce throughput peaks and overload periods. A desirable feature is a workload that tests peaks and overloads while still ensuring the repeatability of measurements.

The transaction mix refers to the percentage of each transaction type that is currently executing on the system or has been executed over the run-time. The proposal should have an adjustable transaction mix that is consistent and representative of the proposed business model.

The transaction rate is the number of transactions per unit of time. A transaction’s rate is affected by an adjustment to the transaction mix, but should also be adjustable independently.

Database Size

Database size within the context of this PDG refers to the concept of how extensively populated the various tables of the OLTP workload database must be to effectively stress the components of the SUT targeted by the proposed OLTP benchmark. And it should also be representative of the real-world configurations modeled by the proposed workload. Ideally, OLTP consumers of the proposed benchmark results would look at the target schema complexity, static and dynamic table cardinalities (i.e., database size) and disk requirements of a benchmark publication and be able to correlate the benchmark performance to their workloads' performance.

Choosing an appropriate size of a database population in a benchmark proposal by necessity consists of a compromise among a number of subject areas. Database size is directly related to schema complexity (see database complexity requirements). Complex schemas may well include some tables with sizable per-row space requirements. When these per-row costs are multiplied by linear scaling of table cardinalities with performance (for some subset of the workload tables), the resulting database sizes can become quite large.

The database complexity section outlines the following requirement: "*Provide sufficiently complex data relationships which encourage alternate data access*". This means that the table row sizes and cardinalities should be sufficient to exercise the overheads seen by production OLTP databases with respect to referential integrity constraints, indexing costs and benefits, and table replication in clustered environments. Similarly, the database size should be to avoid full caching of the entire working set for current and projected SUT designs thus emulating the actual I/O load intrinsic to production OLTP environments. In TPC discussions, it was noted that the constraints of checkpoint-like database behavior to flush modified data in the target workload may interact with database sizing requirements to help inhibit full working set caching. The submitter of a proposal should include a discussion of why they chose their specific table row sizes and how these sizes interact with table cardinalities to produce realistic database sizes as the workload scales with performance.

In addition to a tie to schema complexity, database size is correlated with the benchmark's scaling strategy (see scalability requirements). The scaling section outlines the following requirement: "*Linear growth of the database, or portions thereof, with the performance growth of the system.*" This means that at least some database tables should be twice as large for performance levels twice as high (e.g., in TPC-C, the cardinality of the orders, order-line, and new-orders tables are linear in projected performance). Expectations placed upon OLTP applications with respect to user expectation of online historical transaction access, real-time error checking, extended data types, etc., are evolving as technology evolves. This evolution naturally impacts database size as that it may change both per-row sizes and table cardinalities. Any benchmark proposal should choose database sizes consistent with the evolving sophistication of production OLTP databases for similar schemas and should not just rely on a traditional OLTP terminal data entry model. Initial database population should be realistically populated for the projected load, (e.g., customer tables, history, orders, tracking tables etc. should be populated with a number of rows consistent and proportional to steady state population at the targeted performance levels). Some tables should grow during the measurement interval in proportion to the target workload performance.

The OLTP workgroup, while defining the PDG requirements for benchmark proposals, extensively discussed several shortcomings in the TPC-C workload that evolved into TPC-C benchmark over the decade of its lifetime. With respect to database size, a key shortcoming of TPC-C is that the linear growths of tables with performance levels made disk requirements grow over time to levels well outside of typical OLTP configurations for today's TPC-C publications[1]. Computational performance, governed by Moore's law in the last decade, has improved at rates substantially higher than that of disk *performance*, which has been constrained by the limitations of moving parts. The TPC is unsure how to resolve this seemingly intractable performance dichotomy but would welcome innovative aspects of a benchmark proposal that address this.

A response to this PDG must incorporate the following components directly concerning database sizing into the proposed workload.

- Database size (static and dynamic) *must* be consistent with benchmark scaling strategy.

- The proposed workload *must* require that a reasonable number of days of online retrieval be priced and/or configured

A response to this PDG should incorporate the following component directly concerning database sizing into the proposed workload.

- Database size (static and dynamic) *should* be reasonably consistent with the modeled real-world OLTP application.

Static database size is defined as the size/cardinality of the data which is populated into the benchmark database during the database load process the cardinality of which remains relatively constant during the course of benchmark execution (i.e., some rows may be deleted and others added but overall the number of rows should remain constant). Some database tables in a proposed PDG should have static size. For at least some key database tables, this static size to be populated and maintained must scale with projected OLTP performance. Similarly, the various database tables must be relatively sized consistent to the modeled application (e.g., if an ATM customer table is statically scaled upward, corresponding historical online transaction tables must be statically scaled in a consistent fashion.) The fundamental goal is to attempt to maintain a balanced, meaningful, and representative database population regardless of projected performance level.

Dynamic database size is defined as the size/cardinality of the data added to the database during benchmark execution. In OLTP workloads, the cardinality of some tables grows with the transaction rate. Any PDG proposal must incorporate transaction profiles that increase the cardinality of key tables linearly with benchmark performance. The tables chosen for dynamic growth must be consistent with the modeled application (e.g., if ATM transaction rates are modeled, transaction tables should grow dynamically). Similarly, OLTP workloads are assuming in increasing amounts of online historical transactional history. Any proposal should account for this online storage of dynamic data.

---

[1] See the market relevance section for a discussion of why encouraging representative OLTP configurations is desirable.

In addition to features required by any submissions, the TPC asks those developing proposals to consider a number of additional concepts and consider incorporating them into their proposals where appropriate.

The benchmark proposer should consider how the disk requirements of the workload will grow as technology evolves. The TPC would be very pleased if an innovative way is found to keep disk requirement growth to levels consistent with production OLTP environments running at similar performance levels both at benchmark release and as technology evolves while at the same time satisfying all required proposal features.

The proposed benchmark should incorporate a strategy for populating test databases that are at the same time both loadable in a highly optimized fashion and representative of the production environment modeled by the benchmark. Either an efficient sample loader or a loading strategy included as part of the proposal would be ideal.

The TPC is open to considering proposals that seamlessly integrate more complex media types into their profiles. The storage requirements of these media types may be more variable than traditional OLTP. As an example, some portions of the data in the target workload might use a compressible, future looking media format for some function such as verification images, signature records, fingerprints, etc.

Database Complexity

Database complexity, for the purposes of this PDG, refers to the technologies and schemas defined to represent the environment being modeled. Criticism of existing OLTP-based benchmarks generally revolves around the overly simplistic nature of the schema. Responses to this PDG must include the definition of a representative database schema of the environment being modeled. The schema proposed should contain a representative number of tables, data relationships, and inter-table relationships.

The actual number of tables in the response should be representative of the environment being modeled. The current TPC-C benchmark requires 9 tables. This is generally considered to be an unrealistic number of tables considering what actual customers use. Responses to this PDG must include more than 9 tables in the schema. Responders must also observe caution when designing the schema. An overriding concern of any response must be ease of benchmarking. Specifying a very large number of tables, though possibly representative, can negatively impact the ease of benchmarking. Responders are urged to maintain realism without sacrificing ease of benchmarking.

The response must also include the data relationships in the database(s). These relationships, or constraints, are used to facilitate entity integrity as well as primary key integrity. Schemas also must include the possibilities of alternate or secondary indices. In addition, responses must address the inter-table relationships. Referential integrity definitions must be included which represent the actual customer interaction with the data. For example, if the proposal allows for the deletion of a customer record, all other records associated with that customer must be deleted or modified.

Responses must include the following data integrity types:

- Entity Integrity

    - Enforces row uniqueness within tables.
- Domain Integrity
    - Defines valid entries for columns of data.
- Referential Integrity
    - Preserves the relationship of data between tables, by restricting actions performed on the Primary and Foreign Keys in a table

In designing the database schema, the following requirements must be met:

- Representative number of tables.
    - The exact number must be greater than nine and is dependent on the model chosen for the PDG response.
- Inclusion of static “fact” tables.
    - For example, Zip Codes, Area Codes, etc…
- Provide sufficiently complex data relationships that encourage alternate data access.
    - These access paths should be representative and provide value to the workload.
- Definition of “views” on the data.
    - Facilitates segmentation of data based on some user criteria.
- Concurrency and locking rules.
    - At least as extensive as defined in the current TPC-C specification.
- Encourage data access methods other than database-stored procedures.
    - Significant portions of the transactions should be implemented via ad-hoc business queries (client-side) rather than in stored procedures (server-side).
- Encourage techniques that require the re-compilation of data access plans.
    - Dynamic queries are techniques that can be exploited to require the database to re-compile a transaction execution plan.
- Definition and exploitation of referential integrity features.

Additional consideration will be given to proposals that include the following optional characteristics:

- Inclusion of diverse data types.
    - For example, signature images, product images, etc…
- User Defined Data Integrity

- Business rules which do not fall into other data integrity categories. Implemented using Rules and Triggers.

- Near-current data structures.
  - Replication/Repositories/Shadow databases
- Decision support like data extractions.
  - This extraction does not include the actual analysis of the data, just the extraction from the "online" system.
- Movement of data to near-online storage.
  - Definition of how many days data should be kept online.
  - Methodology of movement (archival) of the data.
- Journaling of database modifications/deletions
  - Provides an "audit trail" of modifications to sensitive data.
  - This journaling is separate from the database transaction log.
  - Can be part of the transaction logic or a function of the DBMS.

These requirements and optional characteristics are designed to insure that the proposed workloads provide value to both the customers of the benchmark and the developers of hardware and software platforms.

In responding to the database complexity section of this PDG, please keep in mind that creating an overly complex schema can negatively impact the ease of benchmarking. If a schema is too large or complex, it may take an unacceptable amount of time to generate the test databases. In addition, an overly complex, or overly simple schema may also negatively impact issues regarding scalability.

Operational Environment Characteristics

Operational Environment Characteristics are requirements that are not unique to an Online Transaction Processing environment. These requirements, while not required to process an order, read or write data to disk, or communicate with other computers, are part of the growing complexity and infrastructure necessary to help guarantee the security and availability of the computing environment.

Security

IMPORTANT NOTE: In an attempt to make use of existing standards or references, and in order to discuss this topic using vendor neutral language, some concepts and terms in this section (e.g. Subjects, Objects, Policies) originate from the U.S. government's Rainbow Series document "Department Of Defense Trusted Computer System Evaluation Criteria" ("Orange Book"). However, despite this use of "Orange Book" language, this PDG is not looking for proposals to define "C2-like" security requirements. Requirements at this level would be far beyond the scope of the TPC, and would compromise the ease of benchmark implementation and management.

At one time, the concept of computer security was realized through the use of physical elements (e.g. walls, doors with locks, security guards, etc.). However, the advent of "The Internet" brought along with it the advent of the "Cyber-Criminal". As a result, physical security, although still necessary, is no longer sufficient. Today, customers have an ever-increasing need for comprehensive software security to protect their information technology assets.

Software security features receive mainstream use – they are an integral part of common online activities. Furthermore, the implementation and use of these features is not performance neutral. For these reasons, the distillation of software security out of the benchmark is not reasonable. Although the benchmark is not intended to be a rigorous security test, it is critical that any new proposal identifies, makes use of, and tests appropriate software security features, e.g. encryption/decryption, Access Control Lists (ACL's).

Today, firewalls are used to protect the outer boundaries of company intranets, and encryption is used during transmissions on the public Internet. More and more, however, people are becoming concerned with the transmission, storage and access to critical information within the intranet boundaries as well. In order to address these issues, companies develop and implement comprehensive security models to ensure the privacy and safety of sensitive information.

For example, accounts with passwords are used to identify and authenticate individuals. The total information space for the company is broken into natural subsets. Policies are then defined to dictate which individuals have which types of access to which sets of data. Furthermore, protocols are established for how data migrates across user or dataset boundaries. And finally policies are defined for how the data is to be stored at any given location within the company.

Once the comprehensive security model is well understood, the final step is to design meaningful ways of testing, evaluating, and validating the implementation of that model

In order to address these concepts, any proposal brought forth must include the following items.

- Network Security –encryption of all inbound/outbound transmissions of personal or proprietary information. Following the above example, any client's personal information that gets transmitted over a public network should be encrypted. Any company confidential information that gets transmitted over a public network should be encrypted (e.g. if the proposed business model has a B2B component, company confidential information may get transmitted over the public network). Transmission of unclassified, or publicly classified data does not require any encryption.
- Application/Service Security - subject identification/authorization for service/application access (e.g. users must login to create/modify/delete personal information, employees must login to perform administrative tasks, etc.)
- Set of tests to validate correct implementation of security model (akin to ACID tests in TPC-C)
- Comprehensive Security Model
    - Subjects

        - Identification and/or classification of user types (e.g. end-user, employee, administrator, etc.)
        - Requirements for identification/authorization of subjects
    - Objects
        - Organizational model of objects
        - Marking of objects with access control labels
    - Policy – rules governing interactions between subject and objects

    Bear in mind that this PDG is **not** looking for C2 or higher levels of security (see Note above). However this PDG **is** looking to grow beyond security levels in current TPC benchmarks. For example, in current TPC-C implementations all users connect to the SUT application using the same username/password pair. Hundreds of thousands of users all connecting using the same username/password pair is too simplistic a security model. Does this mean that each individual should have a unique username/password? Not necessarily. Setting up and managing hundreds of thousands of username/password pairs may make the benchmark too awkward or cumbersome to implement. Proposals should consider the tradeoffs between the complexity of the security model, and the ease with which the benchmark can be implemented and managed.

    As an example, a proposal may define a business environment that has the following high-level security model.

        - *General Public. This class of user has read-only access to all unclassified, or publicly classified, information. No username/password authentication required.*
        - *Clients. This class of user has a defined relationship with the business. In addition to General Public access, Clients can also read and/or write certain individual personal information maintained by the business. For example, clients can review and update their addresses; they can read, but not update, account balance information. Username/password required to access individual personal information*
        - *General Employees. This class of user has access to all public information as well as all departmental information. For example, Sales employees have access to all sales related information using a department level username/password. Customer Management employees have access to all client information using a department level username/password.*
        - *Administrators. This class user has access to business application management functions. An administrative username/password pair is required for access.*
        - *Executives. This class of user has access to internal business management information and functionality. An executive username/password is required.*

In addition to the above requirements, the following list of issues represents possible ways for proposals to further enhance their security offerings.

- Accountability – making use of the ability to trace actions back to responsible subjects.

- Complex Security Model. A complex security model would be one that breaks the proposed business model into multiple departments and defines departmental roles and responsibilities. For example, a business model may include Sales, Marketing, Billing, and Shipping departments. Each of these departments may then have different types of access to different portions of a client's information.
- Encryption of internally transmitted personal/proprietary information
- Formal use or definition of common criteria protection profiles
- On-media encryption of key sensitive data
- Testing/validation of security implementation during measurement interval

Availability

Availability in this PDG is used to require an increase in the data, system, and application availability of the OLTP computing environment over current TPC workloads and requirements. There are some minimum requirements that must be incorporated into a proposal:

- Online disk media recovery
- Hardware failure detection and notification
- Performance loss determination.

These are the first steps to improving the availability requirements in future TPC workloads. Additional concepts to be considered by any proposal are:

- Availability of system resources
- Online upgrade or maintenance of software
- Multiple tiers of availability

Recently the computer industry has begun defining availability in terms of percentage of uptime or 9's availability. If an environment is available 99.99% of the time, it has a four nines availability rating. This rating is a complex calculation involving statistics and MTBF (Mean Time Between Failure). This is probably too complex for the TPC to undertake in its workloads. Instead any new proposal should provide sample tests to ensure compliance with the requirements.

The first requirement is for online disk media recovery. The intention is to not rely solely upon the backup/restore technologies or operator intervention for the protection of data on disk media. Any disk media upon which the database manager, database tables, operating system, boot device or application code is stored should be protected from a single disk failure without relying upon "roll-forward" recovery from log files or restoration from backups. Basically this means that if a disk media fails, the environment will be able to access the data, applications and process transactions without restoring from backup and rolling-forward any changes from the appropriate log files. A simple example of how this could be implemented in today's environment is a RAID-5 disk array with a hot spare. Upon detection of a single drive failure, the disk array re-builds using the hot spare or drive replaced.. The system is accessible at all times while the array works to restore the level of protection. The intention is that a disk media failure should not be a cause for interruption of transaction processing and/or data availability.

The online recovery from a disk failure is only useful when combined with hardware failure detection and notification. This requirement is meant to have the necessary triggers in place to be able to notify operators quickly and conveniently when a fault is detected. The fault detection and notification is not limited to just disk failures, but rather is required for all major components of the system. The requirement should go beyond putting an entry into an error log and displaying a light on the failing component. Rather the notification should be structured as to notify an operator of the failure within 5 minutes of the detection, the location or address of the failed component. For example, if a disk media fails and hot sparing is enabled, then the message should have the failing component, the progress of the hot spare operation and the address or location of the hot spare. Once the hot spare operation is complete and the array back to its original protection level, a message should be sent to the operator of this status and the status and any remaining hot spares. The same requirements should apply also to other component failures.

Many OLTP environments require for the computing environment to be available to process transactions at all times. Maintaining system and application availability is an additional step beyond disk media recovery in that it entails the entire system to have increased levels of availability. No TPC workload currently requires this, so there is an opportunity for new technologies and a learning curve to understand, define and incorporate new requirements. The intent of the requirement is to encourage the development and use of automatic recovery mechanisms in the event of failure of system components. The possible system components that could be tested include memory, CPU's, I/O boards, etc. Tests could be incorporated into the proposal to introduce component failures and measure their impact. These tests should be designed along the same lines as the Durability tests currently implemented in TPC-C. Obviously not every system is designed for 100% availability and cost and market positioning will dictate the levels of availability that are designed into platforms. So along with this requirement is the introduction of the performance loss metric. The performance loss metric will be explained in more detail below.

Measuring and reporting the performance lost is a new secondary metric. The idea is basically to compare the performance of the computing environment after a failure has occurred to the peak performance before failure. Today, computer manufacturers use statistics and mean time between failures to estimate the availability of an environment, but with a standard workload and requirements, the actual impact to the environment can be measured, reported and even used in marketing. The intent is to encourage improvements in the features and levels of availability while making the technology affordable, reliable and comparable. *Following is an example of how this can be implemented:*

1. *Establish peak performance measurement*
2. *Start transactions until steady state is reached*
3. *Induce failure upon component. Note failure time.*

4. *Measure impact of failure upon performance. If processing of transactions is halted because of failure, the performance loss is 100%.*
5. *Restore environment to pre-failure condition and performance levels.*
6. *Report the time from failure until restoration of performance at the performance loss level.*

Another important, but not a required feature to be considered for incorporation into a proposal is online upgrades/maintenance of software components. This feature, while not critical to maintaining the availability of an environment, can decrease greatly the amount of scheduled down time to perform maintenance.

These are just a few of the features that affect availability of OLTP environments. They begin to address some real concerns and are being encouraged to focus development of technologies to improve application, data and system availability.

Online Backup

Whether to provide safety, by complementing or replacing hardware solutions, or reliability, by giving the opportunity to return to a “known” state in minimum time, backups have always been part of any real-world operation. Backups are sometimes also used to extract information from the OLTP database(s) in order to feed offline Data Mining environments.

Yet backup performance has somewhat been neglected in the past, given the opportunity of executing them during some ‘off-line’ time window. But the on-going trend of businesses moving from 8x5 to 24x7 has rendered mandatory the ability to execute backups while processing takes place with yet adequate level of performance.

Online backups have become a critical component of OLTP environment.

A response to this PDG must insure that:

- The workload provides opportunities for executing online backups, possibly incremental ones,
- There is no restriction to- or specification of- a particular backup strategy,
- The emphasis is put on the measured impact of the backup on OLTP performance (reported via a performance loss metric).

Legal

TPC is a standard-setting organization. To enforce the standards that it adopts, it must have the right to control the benchmark. Therefore, TPC will require a nonexclusive license to the specifications proposed in response to this PDG. The attached form of Grant of Rights must be submitted together with the proposed specification.

Where to address questions and responses

Questions and benchmark proposals should be sent to the TPC Steering Committee at: sc@tpc.org.

GRANT OF RIGHTS

The Grant of Rights is between Transaction Processing Performance Council, Inc. ("TPC") and the Company named below ("Author"). Author is submitting to TPC, in response to the Requirements Document dated ____________, a specification, attached as appendix A, for a proposed performance benchmark (the "Work"); the parties wish to establish rights in the Work, and derivative works. In consideration of the covenants and premises set forth herein, the parties agree as follows:

1. **Grant of Rights**.

1.1 Author grants to the TPC, a nonexclusive, worldwide, transferable, fully-paid, perpetual, license to the Work, with the right to create derivative works, including but not limited to all rights of copyright (including any renewals, extensions or revivals of copyright, together with all rights enjoyed by an author under the laws of the U.S. and foreign countries), the right to grant any part or all of these rights to third parties, and the right to use the title of the Work (if any). Any derivative works created by a party, shall be the sole property of the party creating such derivate work.

1.2 The TPC may register the copyright in its derivative works of the Work at its expense in the U.S. Copyright Office. Author agrees, if requested, to execute any documents reasonably necessary for TPC to establish its rights in the Work or any derivative work, and authorizes the TPC to execute and file any other documents on the Author's behalf for purposes of recording or protecting the rights granted to the TPC under this Agreement.

2. **Warranties**. The Author warrants and represents that the Author has the full power and authority to enter into this Agreement and grant the rights herein; that the Work is original, except for material in the public domain.

3. **Miscellaneous**

3.1 This Agreement constitutes the entire agreement and understanding of the parties, and supersedes any prior Agreements between the TPC and the Author. No waiver or modification of any provision of this Agreement will be valid unless in writing and signed by both parties. The waiver of any breach or default of any provision of this Agreement will not e deemed a waiver of any subsequent breach or default. If any provision of this Agreement is held to be invalid or unenforceable, the remaining provisions will not be affected.

3.2 This Agreement will be binding upon the successors and assigns of the TPC and the executors, administrators, successors and assigns of the Author.

3.3 This Agreement shall be interpreted and construed in accordance with the laws of the State of California, regardless of the place of its execution or performance, without regard to its rules relating to choice of law. California courts (state and federal), only located in San Francisco, shall have exclusive jurisdiction over any controversy relating to this Agreement.

Effective date: _______________

TPC

Transaction Processing Performance Council, Inc.

By: ____________________________________

Title: ___________________________________

AUTHOR

____

By: ____________________________________

Title: __________________________________