Benchmarking With Your Head in the Cloud

Karl Huppler

IBM MS XQK
3605 Highway 52 North
Rochester, MN 55901 USA
huppler@us.ibm.com

Abstract. Recent advances in Cloud Computing present challenges to those who create and manage performance benchmarks. A performance benchmark tends to rely on physical consistency – a known hardware configuration, a known software configuration and consistent measurements from run to run. These aspects are not typically present in Cloud Computing. Other aspects change, too. For the consumer, the computation of Total Cost of Ownership shifts to a computation of ongoing expense. Concepts of service and reliability also change from the end-user perspective.

For an organization like the Transaction Processing Performance Council, the expansion of clouds into the commercial, run-your-business space presents new challenges that must be addressed if viable benchmarks are to be created in this important sector of the computing industry. This paper explores these challenges and proposes methods for addressing them.

Keywords: Cloud Computing, Price/Performance, TCO, Total Cost of Ownership, Benchmark, TPC

1 Introduction

Before one creates a benchmark to measure performance of a cloud, it might be reasonable to ask the question “Just what is cloud computing?” The first answer might be “Cloud computing is the use of shared resources that are physically outside of your control to accomplish your work. You are allocated a specific amount of resource from the larger pool of resources available and your work is tracked to see how much resource you actually use.”

This answer appears to be very reasonable, until the questioner responds,

- “How is that different from when, in 1974 I submitted my deck of cards to the operations desk for the University of Wisconsin Univac 1108, knowing that I had to complete my assignments within a specific allocation of compute time?”
- Or perhaps the reply would be “How does that differ from the Service Bureau I worked for in the early 80’s, where the business rented a mainframe and customers would contract for time to run their weekly inventory control jobs?”
- Or perhaps the reply question would be “How does that differ from the PROFS application that I ran in the late ‘80s from my 3270 display running a VM on the System390, appearing as if I had a private computer but being fully aware that I was sharing resources with hundreds of others running similar applications?”
- Or perhaps, “How does that differ from the concepts of ‘The Grid’ that I heard about a decade ago, where computing resources would be treated the same way that electrical and telephone technologies are, with users unaware of where their actual compute resources are located, but know that they will be billed on the basis of the amount of resources that they consume?”
- Or the simple question might be “Isn’t that the same as what we use the web for?”

To all of these questions, the answer is “All of these concepts are embodied in cloud computing.” Cloud computing might be considered to be the natural progression of technology that enables end users or even whole corporations to apply shared resources for dedicated purposes. Perhaps what differentiates cloud computing from prior shared resource solutions is that the progression of technology in hardware, firmware and software enables the shared resources to be distant, reapportioned on the fly and migrated from physical resource to physical resource – in ways that are transparent to the actual user and using methods that are stable enough that the option is viewed as a cost-effective alternative to the use of dedicated resources to accomplish computing tasks.

For performance benchmarks, the growing importance of cloud computing poses some difficult challenges. Most performance benchmarks focus on determining, for some business model, the capacity of a whole system to accomplish work. In the

Full System

cloud, what is important is the capability of a fractional subsystem to accomplish work. Traditional performance benchmarks often have a set of functional criteria that must be satisfied to qualify for benchmark publication, including such things as the ACID properties used to define transactional integrity requirements for TPC benchmarks. These qualities are important in cloud environments, but are often defined in different terms. Particular to the TPC, the benchmarks assume the price quoted is for the complete purchase of all hardware and all software licenses, with maintenance payments over some period of time. Although this could apply to private clouds, in public clouds the concept of making an initial capital investment is in direct contrast to the cloud model.

Fractional Subsystem

Of course, for all benchmarks it is important to establish a business model for which the benchmark will be targeted. One could no more say "I'm going to create a benchmark to represent all cloud computing" than "I'm going to create a benchmark to represent all multiprocessor computing". It must be clear that the benchmark is addressing only a specific slice of the overall cloud computing pie, with a clear description of the application or computing model that the benchmark hopes to represent.

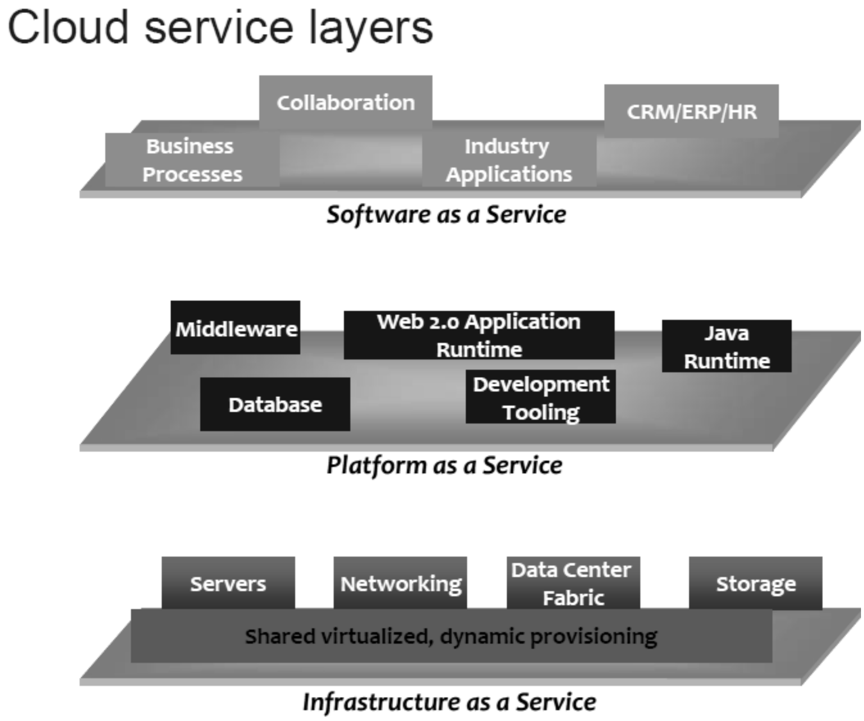
This paper will explore

- The selection process for determining the physical and logical environments that can be measured in a cloud-oriented benchmark
- The requirements for selection of a valid business model and how this may differ from a traditional system benchmark
- The performance aspects that are of particular interest to cloud users, in comparison to those of a traditional, dedicated system
- Features of the benchmark definition that will need to change from those currently employed by the TPC, if a TPC benchmark is to represent a cloud environment.

2 Selection of Cloud Subset for a Target Benchmark

Before designing a benchmark for cloud computing, we must recognize that the space that is "cloud" is massive, and that the benchmark cannot be expected to cover the entire space. Furthermore, there are some areas where performance is a critical component of the compute model and others where it is not. Finally, there are some practical limitations to what can and cannot be included in a general purpose benchmark.

Consider, first, the three primary delivery models services provided by various cloud solutions, as defined by NIST and summarized in the reference materials [1 - 3]:



- **Software as a Service (SaaS):** The entire compute stack is embodied within the cloud. What the end-user sees is that they have access to an application, where the only thing they control is the information that is to be processed. This delivery model is frequently used by individuals for personal needs. Many web-based applications such as photo editors, photo sharing systems and social media applications can be thought of as SaaS applications. Web-based tax preparation applications also fit, here. As the industry matures it is clearly capable of satisfying general business requirements, as well. Certainly, the applications used in a SaaS delivery model could be "benchmarked" for specific tasks – to compare one established cloud offering with another. A web search for "cloud" and "benchmark" typically finds single user tests for this purpose. Such performance characteristics could play a big role in the selection of a service to use. However, in a general purpose benchmark, the application is an integral part of the benchmark and its controls. For this reason, we eliminate SaaS for consideration in this paper.
- **Platform as a Service (PaaS):** Here, the application to be executed is generally owned and controlled by the consumer of the cloud resources, while the rest of the stack is typically owned and controlled by the cloud service provider. For example, http services, database services and

collaborative web application services might be provided by the service supplier – as well as the underlying hardware to support them. The consumer would purchase or author the actual logic that exercises these services – providing a unique web ordering system, or an analysis of consumer habits based on application usage. On line services that provide business support for cataloguing and charging for consumer web sales is a prominent example. Comparing this to a typical TPC benchmark, the benchmark application is under the control of the benchmark sponsor, while the delivery of hardware, operating system, database and middleware functions are under the control of the cloud provider. At first look, this delivery model appears to be closest to today's benchmarks. The "system under test" is comprised of a hardware platform, running a particular OS, using a particular database, and perhaps also using a particular transaction monitor, Java environment, or J2EE provider. The benchmark is the measure of these system components under the stress of the benchmark application.

- **Infrastructure as a Service (IaaS):** For corporate consumers that are first considering migrating to a cloud solution, this may be the natural choice, since they would carry under their own control all of the aspects of the application and the application support software or middleware that it requires. The business model is similar to outsourcing models of the past decade, except that the supplier determines the physical infrastructure in return for guarantees in a service level agreement. The operating system, physical hardware, and the virtualization and provisioning support needed to provide the shared cloud resources are under the control of the cloud provider. Although the vocabulary of the PaaS delivery model may appear to match a public performance benchmark such as the TPC benchmark, the reality of today's public benchmarks is much closer to IaaS. In today's benchmarks, the hardware configuration, the operating system, and the middleware are all tuned explicitly on behalf of the performance of the benchmark application, which is more likely to occur in an IaaS delivery model than a PaaS one, which must support multiple applications simultaneously.

Thus, the target delivery mechanism for a public benchmark for cloud computing would appear to be IaaS, but it is also likely that results from such a benchmark would be used to promote PaaS solutions. In that regard, consideration should be given to defining ways to bring the benchmark definition closer to the PaaS model. Of course, the specific business application must also be taken into account. Some environments are more applicable to a shared resource solution, while others are restricted by regulation or practical reasons [4].

The physical control of the hardware and software components must also be considered: Is the environment Public, Private, or a Hybrid?

- **Public:** The cloud environments that are most highly publicized are those that are offered from both large and moderate sized companies as a service to individuals and other companies. Delivery of consistently strong performance at the contracted level of service is an important aspect of such an offering. Considering its growing popularity, it would be ideal if a public benchmark could be created to measure performance in a public cloud. There are, however, substantial challenges in defining exactly “what” is being benchmarked, since the traditional benchmark is run against a specific configuration of hardware and software. These difficulties should not dissuade the benchmark creators from targeting this environment, but they will mean that the benchmark definition will differ in what is controlled, what is priced and how various functional properties are guaranteed.

- **Private:** A private cloud is more tangible for the “traditional” benchmark developer. The complete physical configuration, the software and middleware, and the associated tuning of these components are all under the control of the owner of the overall system. The challenge, here, is to actually represent cloud computing, instead of a more traditional single application benchmark. In today’s public benchmark environment, where test sponsors are typically the vendors who would like the public to consider their overall solution in purchase decisions, the private cloud environment is a reasonable fit.
- **Hybrid:** These environments are becoming more and more common, either because a consumer wants to have complete control of a part of the resources used, such as storage, or because the consumer wants to handle most computing needs privately and contract for public resources to meet high demand. However, the very term “hybrid” connotes uniqueness, making it a difficult environment for a public performance benchmark and a much better candidate for consumer-specific benchmarks to evaluate a particular business need. It is conceivable that a consumer could combine a private-cloud-oriented benchmark with a public-cloud-oriented benchmark to achieve this end, but it is the assertion of the author that hybrid environments should not be the target of public performance benchmarks.

To summarize the above, there is a place for benchmarks that target both public and private cloud solutions, but the goals of the benchmarks and the nature of what is measured and priced will be very different between the two.

The following table summarizes the key points in this section:

	SaaS	**PaaS**	**IaaS**
Public	Individual measures possible, not recommended for public benchmark	Excellent target for public benchmark, difficult to define	Reasonable target for public benchmark, but more limited scope than PaaS
Private	Typically not applicable; not recommended for public benchmark	Excellent target for public benchmark, slightly less difficult to define, since there is greater control	Excellent target for public benchmark. Easiest to define. Still some challenges compared to traditional benchmark.

Hybrid	Typically not applicable; not recommended for public benchmark	Not recommended for public benchmark due to uniqueness of each consumer environment	Not recommended for public benchmark due to uniqueness of each consumer environment

3 Business Models and Use Cases for Cloud

The Cloud Computing Use Case Discussion Group [1] has done an excellent job of classifying a variety of use cases and the overall functional requirements associated with them. Each use case is discussed here, with regard to how well it might fit within a public performance benchmark.

3.1 End User to Cloud

This use case is almost always associated with an SaaS delivery model. As mentioned in section 2, this is not thought to be a good model for a general purpose public benchmark, although individual user benchmarks are already available for some applications.

3.2 Enterprise to Cloud to End User

In this scenario, the enterprise likely has an application that is targeted to interact with multiple consumers and enterprise users to generate business for the enterprise. It fits well within either PaaS or IaaS delivery models, and could be a candidate for a public cloud benchmark. It would, however, be difficult to measure, since it would require the simulation of both external and internal users as drivers of the workload. Considering the challenges that would be inherent with the use of a public cloud, this may not be the best candidate as a use case for a general cloud performance benchmark.

3.3 Enterprise to Cloud

Here, the enterprise is essentially outsourcing its internal computing requirements to the cloud. While most likely to fit the IaaS delivery model, this use case could also fit the PaaS model. It has the advantage that it can fit on either a public cloud or a private one. It has the further advantage of being most similar to existing traditional benchmarks. Note, however, that there are many more functional requirements to support this use case in a public cloud than there are for a private cloud, and more for a private cloud than would be required of a single-application server environment. Consequently, while this appears to be a candidate for all of these environments, care should be taken to make sure that results that cross these environments cannot be compared.

3.4 Enterprise to Cloud to Enterprise

Supply chain applications are examples of this use case. Often, these applications are fully automated, so they can be a reasonable use case for a benchmark that measures performance in a public cloud.

3.5 Private Cloud

As discussed above, this can be viewed as a subset of the Enterprise to Cloud use case. There are fewer functional requirements required to maintain user security and business integrity, since many of these controls are inherent within the confines of the enterprise. As such benchmarks in a private cloud should not be compared with public cloud environments. It would not be a fair representation to compare the two different sets of functional requirements, nor would it be representative to require the private cloud to rise to the requirements of the public one.

3.6 Hybrid Cloud

This use case is interesting in the number of functional requirements needed to maintain integrity when some resources are private and some are public. However, as discussed in Section 2, it is difficult to conceiver of a “general purpose” definition for a hybrid cloud, so this case is not recommended for the definition of a public performance benchmark.

4 Performance Criteria Important to Cloud Environments

Performance criteria such as response times and throughput are important in any computing environment – or they become important when they fail to meet expectations. However, there is much more to “business performance” than the measure of transaction response times and overall throughput. This becomes particularly important in a cloud environment, where the resources being used are not under the direct control of the enterprise and where the physical resources being exercised may actually change from day to day, or even moment to moment. Areas of reliability, consistency of service, and the ability to expand and contract allocated resources all play a role in the overall business performance that is delivered by the solution. These areas are important in a dedicated environment, but become key purchase criteria in shared resource environments such as cloud computing.

The following table is adapted from an IBM article on performance considerations for cloud computing [3]. The table was originally used as a list of performance-related items that might be included in a Service Level Agreement. The table has been adjusted by replacing the columns associated with an SLA with the last column that contains comments on the applicability for use as a measure in a general purpose performance benchmark.

Service Level category	**Key Performance Indicator**	**Applicability to a public performance benchmark for cloud computing**
Availability	Percentage of time that service or system is available	For public clouds, should be stipulated in the minimum SLA requirements, with transparent or near-transparent fail-over for resources in error and guarantee of data integrity across the migration
	MTBF - Mean time between failure	Covered above
	MTTR - Mean time to repair	Covered above
Performance	Response time for composite or atomic service	Particularly important to set strong criteria for response times in an environment where the physical location of compute resources are not known
	Completion time for a batch or background task	Same as response time, only for long running "transactions"
	Throughput - Number of transactions or requests processed per specified unit of time	Takes on new meaning for cloud. Assumption is that there is always additional resource available to obtain more throughput, so the measure is more likely to be throughput per unit of resource in the SLA
Capacity	Bandwidth of the connection supporting a service	For public clouds, likely a part of the price equation, but should also be a part of the minimum SLA
	Processor speed – Clock-speed of a processor (CPU)	Reported in the benchmark and a contracted part of an SLA. Perhaps a component in the overall throughput measure
	Storage capacity of a temporary or persistent storage medium, such as RAM, SAN, disk, tape	Part of price equation and part of an SLA
Reliability	Probability that service or system is working flawlessly over time	In today's environment, this should be covered by the availability requirement, above
Scalability	Degree to which the service or system is capable of supporting a defined growth scenario	Although a benchmark may be run on a fairly static set of resources, the ability to scale to larger resource use or even to scale to smaller use is a trend that should be included, perhaps even as a dynamic part of the benchmark.

5 Benchmark Requirements for the TPC

For the Transaction Processing Performance Council, there are some special considerations that must be made in the definition of a benchmark that is focused on cloud computing. None of these challenges are insurmountable, and the addressing of them can produce a much richer benchmark than if they are ignored.

5.1 Price

The TPC's Policies document requires that each TPC benchmark have three primary metrics: a performance capacity metric, a price/performance metric, and an availability date. The rules for generating a price to use as the numerator of the price/performance metric are defined in the TPC's Price Specification and the associated TPC Benchmark specifications [5 - 8] and can be summarized as follows:

- Purchase price of the complete hardware configuration
- Purchase price of licenses for software needed to develop and execute the benchmark application
- Price to maintain the hardware and software for a period of three years

For price considerations, we find a significant difference between public and private cloud solutions. The two pricing models are different enough that it may be difficult to achieve comparability between them.

From a benchmark-ability perspective, there are some advantages to using a private cloud model – in that all of the resources are under the control of the benchmark sponsor. However, from a price perspective, and also from a benchmark execution perspective, the need to demonstrate that the benchmark application is only using a fraction of system resources and should only be charged for a fraction of system costs will be a significant challenge.

5.1.1 Pricing Public Cloud Configurations

Particularly for a public cloud, one of the reasons a consumer opts to contract for services is to replace the surge-cost of the purchase of a system that likely has more capacity than is typically needed with more cost-effective expense costs of ongoing monthly service fees for the cloud resources. Instead of up-front payment of the capital equipment costs of the entire configuration, the consumer pays through some combination of three methods that are in contrast to the TPC Pricing Specification's requirements:

- Regular monthly fee for contracted resources, such as X compute cores with Y memory and Z storage ("rental" model)
- Monthly charge for resources used, such as $XX/100GB transfer ("utility" model)
- Fee for specific resource allocation, such as $YY for 25,000 core-seconds of compute time (" prepaid phone" model)

In addition to replacing the items required in current TPC prices (and depending on the service-level-agreement negotiated with the provider), the monthly service fees typically include many of the items that are a part of the overall cost of ownership of compute resources, but are not included in the TPC's price requirements [10]. For public solutions, key items include (some apply to PaaS but not IaaS delivery models):

- Transparent hardware upgrades
- Transparent middleware upgrades
- Database and Middleware administration
- Operational Support
- Electricity
- Floor space and other building costs
- Backup and Recovery services
- Up-time guarantee and associated migration services

By shifting to a 3-year expense cost model for public cloud environments, instead of a purchase + maintenance model, the TPC could accommodate typical cloud pricing models and include these key areas of the total cost of computing that are currently missing in TPC prices. While there are other aspects of benchmarking in public clouds that would be more difficult, this enhancement to pricing requirements could be accomplished by specifying the minimum support required in the contracted Service Level Agreement to ensure that all items are included.

5.1.1 Pricing Private Cloud Configurations

For private cloud configurations, the inclusion of the original TPC price list is possible, but it should be altered to reflect the inclusion above list of items that are missing from current TPC prices and to reflect the fact that cloud solutions are designed to use only a fraction of the total computing power that is available. Including the SLA list from above is more difficult for a private cloud than a public one. For the public configuration, the supplier has completed their own assessment of the collection of costs and has rolled them into the service fee, usually without itemization. For the private cloud, an actual consumer would accomplish something similar, but benchmark rules for establishing a uniform methodology will be a challenge - because it requires some assessment of the cost for administrative and operational support, or for building requirements, which are difficult to define and can change from locale to locale. However, software upgrade support and electricity can certainly be included, as can some requirement for the hardware and software necessary to support fail-over and server migration.

The greater challenge for private cloud configurations is that the price of the configuration in TPC terms is for the entire configuration; as if a single application is using the entire set of resources. However, the same cost savings that attract consumers to public clouds are also what attract them to private ones - - Individual applications do not absorb the entire configuration, so the users of an application get a charge-back only for the resources that they consume.

As a starting point, assume that the total configuration is, on average, used only at 2/3 of the total resources for all possible applications, to ensure sufficient head room for expansion while maintaining appropriate quality of service. Then, a method must be devised to assess what fraction of the configuration the benchmark application is consuming. The ratio of this fraction to 2/3 can be applied to the system costs to derive a benchmark cost. If the total configuration cost is computed using a lease model instead of a purchase model, this method can approximate the method proposed for public clouds. As with public clouds, it will be important to stipulate a minimum set of requirements from the SLA list, above, to ensure that the configuration is a “true” cloud.

5.2 Availability

The TPC’s rules for the availability of hardware and software components are also defined in the TPC Pricing Specification. Essentially, the requirement is that all components required for the benchmark be publicly available for delivery within 185 days of the publication of the benchmark result.

For benchmarks that are run on public clouds, the assumption is that the availability date is the date of the benchmark execution, since it is using publicly contracted resources. For private clouds, the TPC’s existing requirements can stand as they are.

5.3 Energy

The TPC’s Energy Specification [9] has a set of rules to be followed for generating the optional energy metrics for each of the existing TPC benchmarks. The goal is to promote these rules to apply to future TPC benchmarks, as well.

As mentioned in the price section, above, for public clouds, the cost of the energy consumed to support the fractional set of resources in the contract is included in the overall monthly fee that is charged. Although the host of the public cloud certainly has to pay attention to energy consumption, it is not a concern for the consumer and one assumes that the host is providing a price quotation that allows the continuation of payment of these and other costs. Thus, the measure of energy for a public cloud benchmark is likely not required.

For a private cloud, the measure of energy is achievable and likely very important. The challenge is reflective of that described for pricing of private clouds – Assuming the benchmark application is measured on a fraction of the total physical resources available. How does one determine how much of the overall energy consumption should be allocated to the workload? Clearly, the most efficient use of energy is if the system is operating at capacity, but if the benchmark application is using the entire system, it isn’t really a cloud. Some rule could be established to observe the fraction of the total capacity that the benchmark application is using and apply some formula to that fraction to arrive at a reasonable energy value.

5.4 ACID Requirements

All TPC benchmarks have data integrity requirements for Atomicity, Consistency, Isolation and Durability. These requirements can and should stand for cloud computing environments, as well – both public and private environments. The nature of the individual tests to ensure these properties may need to be altered.

In the case of Durability, the overall requirement should be altered. As noted earlier, one of the likely requirements for a Service Level Agreement for cloud resources is a guarantee of up-time and migration of services to maintain that up-time in the event of a failure. Thus, the durability requirement for a cloud computing benchmark should not only require that data integrity be maintained after recovery from a failure, but that data integrity be maintained while the application environment is migrated to new physical resources to simulate the loss of the original resources. The migration time and the relative reduction in throughput during and after the migration should be measured and reported as a part of the benchmark metrics.

5.5 System Capacity and Quality of Service (QOS)

TPC benchmarks (and most other industry benchmarks) focus on the capacity of a total configuration to accomplish work under a particular load. If 100% of the system resources are not used to accomplish this, the benchmark sponsor either finds a way to increase the load to use 100% or reduces the scope of resources configured so that 100% are used. Cloud computing is based on the premise that the application will never use 100% of the total system resources – but rather a fraction of the total resource available.

While the overall throughput achievable for the contracted resources is an important measure, there are other performance criteria listed in Section 4 of this article that are also important. Many of them are important in a single-application environment, as well, but become more prominent in cloud computing. Some TPC benchmarks require running within a maximum response time limit, but these requirements are often quite relaxed. In a cloud benchmark, not only the response time component of QOS is important, but overall bandwidth, system availability, resource migration time, resource-on-demand time, and other aspects discussed in section 4 are also important.

TPC Policies require that there be a performance metric for the benchmark, but do not stipulate that it be solely a throughput metric. For cloud computing environments, it is reasonable and advisable to include other components of performance in the overall performance score of the benchmark.

6 Summary Points

Cloud computing is here, and growing. Performance considerations for applications within a cloud environment are an important part of the selection and implementation of a solution, just as they are with a single-application or single-server environment. However, the specific performance criteria for cloud are sometimes different or sometimes treated differently than they are in a more traditional, stand-alone computing environment.

Benchmarking in "the cloud" is desirable and achievable, but is not trivial. To define a public performance benchmark for cloud computing requires changing from a total-system-total-capacity benchmark process to a partial system mind-set in which many of the performance measures will be different than they are with a total system measure.

As with any benchmark, it is important to select an appropriate use case, or business model to target the benchmark towards, and a delivery model that is germane to that business model.

Use cases that are most likely associated with a Software as a Service (SaaS) delivery model are more likely the topic of individual user environments and are not recommended for more general public performance benchmarks. Use cases that fit the Platform as a Service (PaaS) delivery model can be good cases for measures of public clouds. Use cases that fit the Infrastructure as a Service (IaaS) delivery model are more likely candidates for measures of private clouds. Of the variety of use cases that have been defined, the "Enterprise to Cloud" use case is the most likely candidate for an initial attempt at creating a cloud benchmark. There are sufficient differences between public and private clouds that a benchmark should not attempt to span both.

For the TPC, the inclusion of cloud computing in benchmark designs will require rethinking how performance and price are measured, and the way that these metrics are represented to the consumer. A critical component of any cloud benchmark will be the inclusion of minimum service thresholds that will be typical of Service Level Agreements that will be established to bring enterprise solutions to the cloud space.

References:

1. Cloud Computing Use Case Discussion Group; Cloud Computing Use Cases White Paper – Version 4, 2010, http://opencloudmanifesto.org/Cloud_Computing_Use_Cases_Whitepaper-4_0.pdf
2. Oracle – Anonymous; Oracle Cloud Computing; 2011, http://www.oracle.com/us/dm/oracle-cloud-computing-final2-332097.pdf
3. Duijvestijn, Ferrnandes, Isom, Jewell, Jowett, Stahl, Stockslager; Performance Implications of Cloud Computing, 2010, http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101684
4. IBM – anonymous; Dispelling the vapor around cloud computing; 2010, http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101684

5. Transaction Processing Performance Council – Anonymous; TPC Pricing Specification Version 1.6.0, 2010, http://www.tpc.org/pricing/spec/Price_V1.6.0.pdf
6. Transaction Processing Performance Council – Anonymous; TPC Benchmark C Specification Version 5.11.0, 2010, http://www.tpc.org/tpcc/spec/tpcc_current.pdf
7. Transaction Processing Performance Council – Anonymous; TPC Benchmark E Specification Version 1.12.0, 2010, http://www.tpc.org/tpce/spec/v1.12.0/TPCE-v1.12.0.pdf
8. Transaction Processing Performance Council – Anonymous; TPC Benchmark H Specification Version 2.14.2, 2011, http://www.tpc.org/tpch/spec/tpch2.14.2.pdf
9. Transaction Processing Performance Council – Anonymous; TPC Energy Specification Version 1.2.0, 2010, http://www.tpc.org/tpc_energy/spec/TPC-Energy_Specification_1.2.0.pdf
10. Huppler; Price and the TPC; TPCTC 2010; LNCS Vol. 6417 pp 73-84; Springer Heidelberg 2010