



# **IDC QPI**

Qualified Performance Indicator

## **Analysis & Methodology**



# EXECUTIVE SUMMARY

IDC's Server Decision Suite provides IT buyers and sellers with a common set of metrics to compare servers of different architecture, age, and brand. In support of this suite, IDC created a system performance methodology that combines system configurations and industry-recognized benchmark results into a model of relative and normalized server performance indicators.

Hardware characteristics, such as server architectures, processor chips, cores, and main memory, as well as results from selected standard benchmarks are collected and processed using a tool, named the QPI-Engine™ to generate Qualified Performance Indicators (QPI) for current and legacy servers from major system vendors. IDC commissioned InfoSizing ([www.sizing.com](http://www.sizing.com)), a consulting firm with proven leadership in the field of information systems performance measurements, to assist in the design, development, validation and continued maintenance of this tool.

This paper starts with an analysis of the issues inherent in the creation of an analytical model geared toward the generation of performance estimates. It then differentiates the concept of performance estimates from the concept of a Qualified Performance Indicator. The paper concludes with an outline of the main design principles and the architecture at the core of the QPI-Engine™.

# PERFORMANCE MODELING

## Measuring Performance

Performance plays a crucial role in the development, deployment, and operation of an information processing system. During the development phase, a good understanding of the performance characteristics of the various implementation options can have a strong influence on design choices. Without properly sizing each computer system on which the solution is being deployed, the success of the project will likely be compromised. And as the demands on the deployed solution evolve over time, accurate planning for additional capacity cannot be conducted without reliable performance information.

Because all systems operate in an application environment characterized by a specific workload, their performance is tightly related to that workload. The level of performance of a system under a given workload is likely to vary significantly from its level of performance under a different workload. Consequently, to be relevant and accurate, information about a system's performance must be evaluated in the context of its targeted workload.

## Modeling vs. Measuring

While the ideal and most accurate way to obtain performance information about a system is to measure it under the actual load it is meant to carry, that option is seldom available or practical. During the design phase of a project the system is not yet available for testing, and yet, verifying that its capacity can handle the expected load is critical. Once the system is built and in production, disrupting its operation by taking performance measurements may be difficult, or simply not a viable option.

An alternative to taking actual measurements of a system's performance is to build a mathematical model of the system's behavior and to use this model to analyze the performance characteristics of the system. The ideal model would capture all aspects of the system and be able to predict its behavior accurately under any workload. While such an ideal model may be an option for very simple systems, it becomes less of an option as the complexity of the system and its associated workload increases. In the case of information processing systems involving multiple large computers serving a complex and dynamic workload, building such a model is hardly an option.

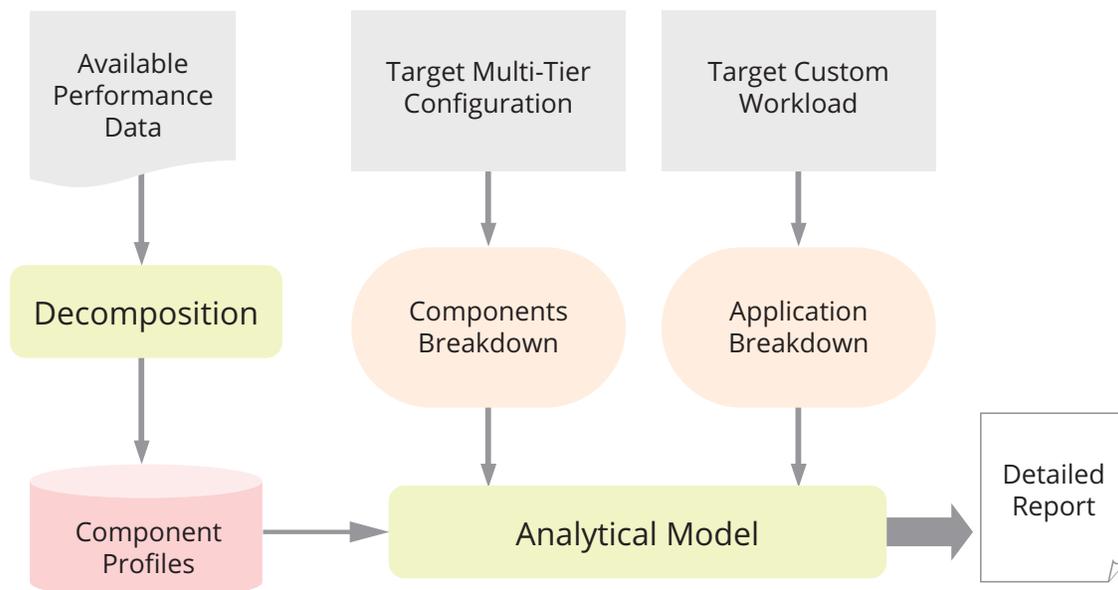
While an exact model of a system’s behavior is rarely a practical option, an approximation of such a model is a possible and desirable alternative. As for all approximations, the usefulness of such a solution is based on the adequacy of its expected margin of error. An approximation for which the margin of error is greater than the precision needed to make good use of the data is mostly misleading and can result in costly design flaws.

Assuming that the need for precision is compatible with the level of approximation delivered by the model, several options should be considered. On one end of the spectrum is a general purpose simulator based on an extensive analytical model. On the other end of the spectrum is a single-purpose simulator based on simplified modeling techniques. Both of these options can be adequate solutions if used in the proper context and with a good understanding of their limitations.

## General Purpose Simulator

A general purpose simulator is a modeling tool designed to analyze the performance characteristics of a complete computing complex under a fully customizable workload. Such a simulator would have the ability to explore multi-tier configurations by breaking them down into their individual components, such as processors, main memory, I/O channels, and storage sub-systems.

On the workload side, a general purpose simulator would accept a detailed definition of the target application, including characteristics such as its type (OLTP, Decision Support, Scientific, etc.), its level of complexity, the data set against which it operates, and the pattern of use for this data. The diagram below illustrates the operation of such a general purpose simulator.



Major system vendors commonly build and use general purpose simulators to advise their customers on how to best configure new systems, expand existing ones, or consolidate multiple application platforms into a single server. These tools use available performance information to produce a capability profile for each system's components. In turn, they produce a detailed report on the trade-offs involved in obtaining the best match between a target workload and a proposed configuration.

## THE SINGLE NUMBER

### Need for a Single Number

While a general purpose simulator has a broad range of applicability, some situations call for a single purpose simulator. Such a simulator uses a simplified version of the modeling tool found at the core of a more complex general purpose simulator. This simplification is made possible by a reduction in the simulator's scope and applicability.

One such situation is the generation of a single performance number that encapsulates the generic performance capabilities of a simple configuration. In this case, the configuration being simulated consists of a single-box server running against a fixed, generic workload. The result of this simplified simulation is not a detailed report, but rather a single, generic performance number. Such a proverbial "one size fits all" performance metric finds its use in many areas, such as requests for proposals (RFP), system pricing or competitive marketing.

The success of standard benchmarks from organizations like SPEC and the TPC is proof that single numbers for single servers are in great demand. In almost every one of these benchmarks, the targeted component is a single system and its performance is ultimately reported as a single metric. It is no coincidence that the few standard benchmarks designed to measure the interaction of multiple systems (such as SPEC-JAppServer or TPC-W) are among the ones with the fewer published results.

There is a clear need to generate a single measure of system performance, normalized across multiple system families from multiple system vendors. These measures exist within the context of a limited application environment. However, if it was feasible, the computer industry and its customers would favor the simplicity of a unique measure of system performance, applicable across a wider range of application environments. This quest for a "single number" is neither new nor is it likely to disappear in the foreseeable future.

## Many Single Numbers

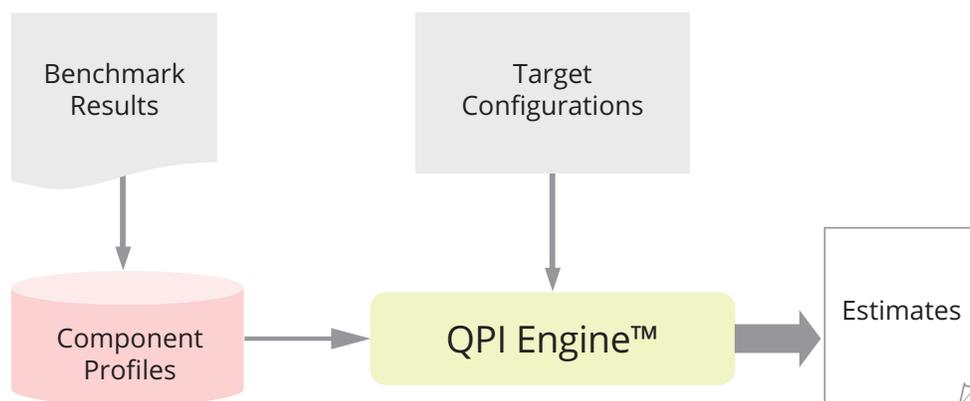
Paradoxically, the “single number” concept comes in many flavors. Each application environment, such as OLTP, Decision Support, Web Services, Financial, or Scientific applications, has its own measure of system performance in the form of its own single number metric.

The market demands drive the definition and the production of each of these performance metrics. Not surprisingly, the needs of the largest markets tend to be served first. System vendors tend to invest more resources in the production of performance information aimed at the commercial data-processing market than for the scientific computing market. Nonetheless, each of these markets is addressed independently and many single numbers are published for every new system announced by vendors.

## Performance Estimates

Using a single-purpose simulator, the performance of simple configurations running simplified, synthetic workloads can be estimated through simulation rather than actual measurements. In such a simulator, the complexity of the estimated configuration is limited to a single-box server attached to a limited number of storage devices. Similarly, the workload for which the system’s performance is estimated must be relatively simple and well understood.

The diagram below outlines the operation of such a single-purpose simulator. The engine producing the performance estimates, here represented as a “black box,” is populated using actual benchmark measurement results, which have been compiled into performance profiles for the various components of the configurations being evaluated. The limited scope of these configurations and the inherent homogeneity of the benchmark results used to calibrate the engine make it possible to build it using simplified modeling techniques.



# QPI-Engine™

## Qualified Performance Indicator

Responding to the demand for a global performance rating for single-box servers and leveraging the potential to use a single-purpose simulator to produce performance estimates, IDC developed the QPI Engine™.

The acronym QPI stands for Qualified Performance Indicator. The name was chosen to capture two of the Engine's underlying concepts. The term "indicator" implies that the performance numbers produced by the Engine are not measurement results obtained from executing an actual benchmark, nor are they calculated estimates of these specific results. Rather, the numbers produced by the Engine are synthetic indications of the performance capabilities associated with the corresponding systems.

Another underlying concept of the QPI-Engine is captured by the term "qualified." It expresses the notion that these performance indicators are to be used within a specific context. As such, IDC qualifies where and when the indicators produced by the Engine should be used and advise against using them outside of this defined context. More specifically, the applicable use is defined as follows:

*The Qualified Performance Indicator (QPI) should be used as a generic source of comparative performance information between single-box servers of similar capabilities when deployed in a commercial data processing environment. Given its generic nature, this indicator is best used to highlight performance differences greater than 20%.*

More specifically, the indicator provides performance information in a generic context, rather than for any specific workload. With this information, systems with comparable capabilities and built around similar technologies can be compared. However, the applicability of the indicator diminishes as the scale of performance between compared systems increases, or as the generational gap in technologies widens.

In addition, given the generic, "one size fits all" nature of the indicator, it is well suited for comparing systems with significant (more than 20%) differences in performance. Its use might lead to misleading conclusions when looking at systems with near-equal capabilities.

## Design Choices

A number of design choices were made for the construction of the QPI-Engine™. One of these choices is the assumption that the server's performance capabilities are fully exercised. In other words, the processors configured in the server have reached maximum effective utilization levels.

Another design choice involves the server's memory configuration. Since the size of the main memory configured in a server has a significant impact on its performance capabilities, the Engine assumes that an optimal memory size was configured. This optimal memory configuration is reported alongside the QPI.

The design of the QPI-Engine™ also relies on multiple sources of information regarding the measured performance of servers. The Engine leverages vendor-supplied performance data that, in most cases, are not generally available. Through a collaborative relationship with system vendors, IDC has access to internal benchmark results and other low-level performance characteristics. When applicable, this information is used to increase the accuracy of the performance indicators generated by the Engine.

Published results from actual performance benchmark are also used by the *QPI-Engine™*. Most of these results are for benchmarks defined by standard organizations like the Standard Performance Evaluation Corporation (SPEC) or the Transaction Processing Performance Council (TPC). Each of these standard benchmarks involves a controlled workload aimed at characterizing a specific application area within the commercial data processing space. Therefore, the relevance of these standard benchmark results and their use by the Engine is carefully weighed.

## Standard Benchmarks

Ideally, the QPI-Engine™ would focus on using performance data collected from real-life applications. In actuality, real-life workloads are fast changing by nature as they adapt to the evolving needs of the business. These frequent changes prevent their use for long-term comparisons.

In contrast, standard workloads remain unchanged over longer periods of time, resulting in the productions of comparable performance results across a wide array of servers. This comparability is what makes standard benchmark results relevant and usable by the Engine.

Results from the following standard benchmarks are used by the QPI-Engine™:

### Active Primary Benchmarks

**TPC-E:** This OLTP benchmark is of medium complexity. It has a growing body of results that have, over time, achieved its intended purpose as replacement for the older TPC-C.

**TPC-H:** This basic data warehouse and decision support workload was released in 1995 under the name TPC-D. Later revised in 1999 under its current name, it provides a large body of comparable performance data on older, legacy servers as well as current ones.

**SPEC CPU:** This benchmark is focused on measuring raw processor power, expressed in terms of performance for integer operations (SPECint\_rate) and floating point operations (SPECfp\_rate). It was first released in 1995 and revised in 2000 and 2006. Currently in use are SPEC-CINT2017rate and SPEC-CFP2017rate.

**VMmark:** This proprietary benchmark is currently used at version 3. It simulates web-scale multi-server virtualization. Designed to measure workload consolidation performance originally. It is peer reviewed and has increased in complexity and applicability with each version update.

**SPECjbb:** This benchmark measures the performance of a Java business application as well as its underlying Java Virtual Machine (JVM). Originally released in 2000, it only gained wide popularity after it was revised in 2005. The current SPECjbb2015 results are used.

**SAP SD:** This proprietary benchmark measures the performance systems running SAP's Enterprise Resource Planning (ERP) and associated applications under a controlled load. It provides a solid body of results, but with limited comparability over time due to the regular revisions of the underlying applications. However, it is the most popular measure of performance in a simulated real-life environment. SAP SD 2-Tier is in use.

### Legacy Server Benchmarks

**SPECweb & SPECjAppServer:** These retired benchmarks measured Java specific application with a database tier and the performance of systems under a workload simulating the activities of a web server. They still provide a body of comparable performance data on older, legacy servers.

**Oracle (eBS):** This proprietary benchmark measures the performance of systems running Oracle's Financial application under a controlled load. Initially released in 2001, it provides a limited body of results with comparability diminished by a few

revisions of the workload over its life time. But it is one of the few available measures of performance in a simulated real-life environment. It still provides some comparable performance data on older, legacy servers.

## Inside the Black-Box

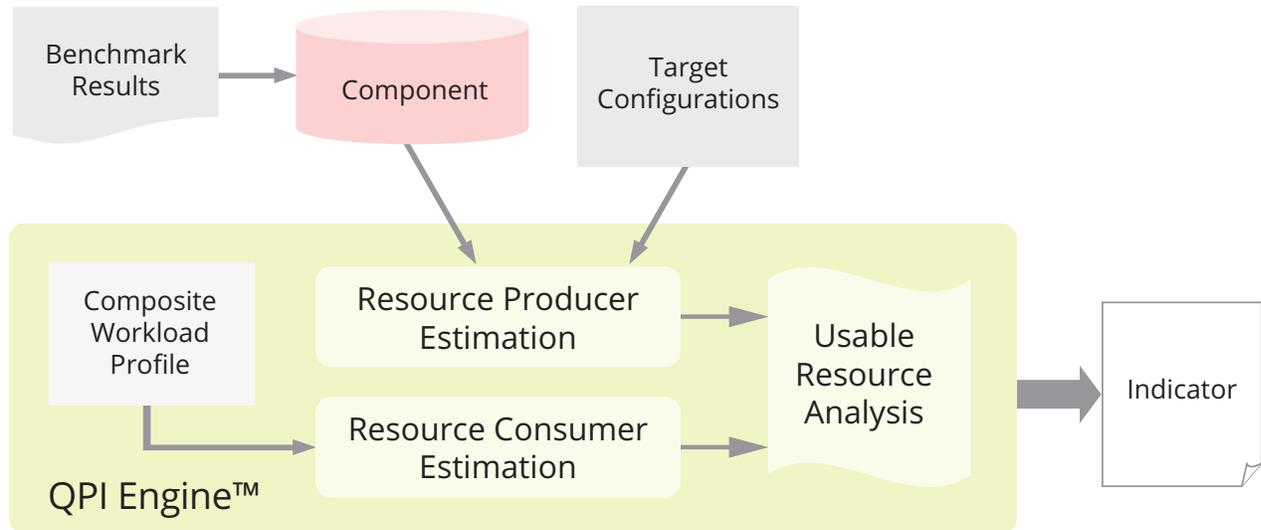
A more detailed description of the QPI-Engine™ involves taking a look inside the “black box” powering the Engine. At the core is a single-purpose simulator operating under a “producer-consumer” paradigm. In a producer-consumer paradigm, the available hardware is the producer of processing resources, while the workload applied to the system is the consumer of these resources.

By analyzing the dynamic between the producer and the consumer of computing resources, the Engine identifies how much of the available hardware is usable by the workload and can contribute to improving the performance capabilities of the targeted system. Other simulation models may assume that all of the components configured in the target system can be fully used, regardless of their ability to improve the execution of a specific workload. In contrast, the producer-consumer model only takes into consideration the portions of configured components that can actually be used to improve performance.

Following are some of the typical scenarios where the producer-consumer model can provide significant benefits and increase the accuracy of the simulator:

- The capacity of a given resource only impacts performance if it is needed to fulfill the demands of the workload. For example, a larger main memory will only improve performance if the workload has a use for the larger size.
- Available resources only contribute to performance if they are not constrained by the limits of other resources. For example, a faster CPU will improve performance unless it is already “starved” by an undersized memory channel.
- An increase in the number of components of a given type only improves performance if the workload can use them in parallel. For example, additional channels in the I/O sub-system will only improve performance if the access pattern is made of a large numbers of small, random I/Os.

The following diagram illustrates the QPI Engine’s framework and its use of the Producer-Consumer model:



To estimate the capacity of the components configured in the targeted system (i.e., the resource producer), general capacity information about each component is combined with factual performance information. This information, derived from benchmark results, is decomposed to focus on individual components.

To estimate the performance demands placed on the targeted system (i.e., the resource consumer), the workloads from the standard benchmarks listed above are combined into a composite workload profile.

In the QPI-Engine™, the producer-consumer model is applied to analyze the performance capabilities of the configured processing resources. It is also used to determine the optimal memory configuration that will allow maximal use of these processing resources.

## Hardware & Measurements

The QPI-Engine™ is built on the principle that the hardware characteristics of a system are the most significant source of performance information. In other words, studying the details of each configured component says more about the performance of the system than any other source of information.

Among the basic hardware components, the processor drives all of the performance capabilities. Other components, such as the main memory or the I/O subsystem, act

as surrogate for the processor. They can allow the processor to make use of all of its capabilities or they can limit these capabilities by creating a bottleneck in the system. Over-scaling the configuration of these components can ensure that they do not get in the way of the processor. However, the cost of over-scaling components such as the main memory is often prohibitive. This is one of the motivations behind using the QPI-Engine™ to determine optimal memory sizes.

When evaluating the performance characteristics of a processor, the QPI-Engine™ takes most aspects of the processor into consideration. These include regularly published characteristics such as clock speed or number of cores and threads. But it also includes subcomponents such as size of the instruction and data cache, size of the second level cache or bandwidth of the memory bus. In the case of some processor architectures, the QPI-Engine™ may also make use of information regarding the location of the memory controller, the coupling of multiple cores or the size and density of the chip's die.

Using the characteristics of the hardware, the Engine can produce a QPI with a first level of accuracy. But to increase its accuracy further, the Engine leverages the analysis of benchmark results. This analysis is essential to uncover the performance information that is not made visible by the examination of the hardware performed by the Engine.

Ultimately, the performance of a system is only relevant and meaningful in the context of its application area and the workload it carries. The broad composite workload profile used by the Engine anchors the application context that qualifies the QPI 's accuracy and its applicability.

# CONCLUSION

IDC developed the QPI-Engine™ to generate server performance indicators. Based on a producer consumer model using the hardware configuration of the servers, and leveraging multiple standard benchmark results, the Engine can be applied to the generation of QPI values for current as well as legacy servers.

The primary use of the generated QPI values is to assist in the selection of comparable server systems as presented in IDC's Server Decision Suite. The values also provide a backbone for measuring the performance of new versus replacement servers that are part of a server migration or consolidation exercise. Used in this context, the QPI-Engine™, combined with the qualification of its area of applicability and the review of the generated QPI by system vendors, provides a relevant, accurate, and independent indicator of server performance.

For more information about the IDC Qualified Performance Indicator (QPI) please contact Brian Clarke at [bclarke@idc.com](mailto:bclarke@idc.com) or call +1 (508) 935-4765 or Ken Hoyt at [khoyt@idc.com](mailto:khoyt@idc.com) or call +1 (508) 935-4238.