

Administración de Recursos

Benchmarking

¿QUÉ ES BENCHMARK? ¿PARA QUÉ SIRVE?

Benchmark es un **proceso sistemático y continuo** que permite evaluar comparativamente los productos, servicios y procesos de trabajo en organizaciones. Estas mediciones son las que van a definirnos cuál es el rendimiento de lo que estamos evaluando en comparación con lo que se está analizando en la medición y así permitir la toma de **decisión** correspondiente. Podemos medir cualquiera sea la cualidad de lo que nos encontramos evaluando siempre y cuando comparemos las mismas características entre todos los postulantes que están siendo sometidos a través de esta técnica.

El benchmark permite:

- *Comparar elementos a través de características claves para la solución:* Se estudian aquellos componentes y/o parámetros que aportan datos relevantes a la decisión que se debe tomar.
- *Obtener resultados objetivos*
- *Obtener la mejor relación costo/beneficio:* toda decisión se basa en gran parte en esta ecuación que balancea el presupuesto existente vs las exigencias y necesidades.
- *Comprobar si los elementos estudiados se adecuan a las necesidades:* los resultados de un benchmark permiten conocer si un elemento estudiado cumple o no con los requerimientos que se tienen y además permite conocer su comportamiento ante determinadas situaciones (Simulaciones).

ETAPAS DEL PROCESO

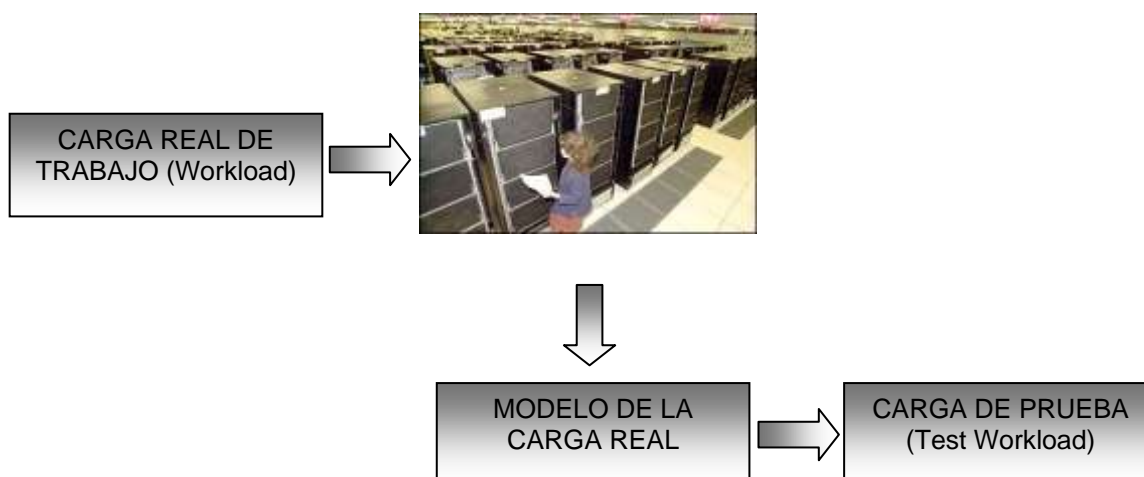
Determinar el elemento de estudio

- Determinar el elemento que se va a someter a estudio
- Elegir los factores y variables claves que se tendrán en cuenta para realizar el benchmark.
- Seleccionar las opciones disponibles en el mercado que se evaluarán en el proceso de benchmark.

Preparar el entorno de prueba

- Recopilar requerimientos del tipo de Benchmark a realizar.
- Realización de tareas previas a la etapa de ejecución. Preparación del ambiente en donde se realizara la prueba. Ejemplo: creación de lotes de prueba o carga de prueba (*test workload*), creación de tablas, etc.

El armado de la carga de prueba no es algo trivial. Para que los lotes utilizados sean representativos debe analizarse la carga real de trabajo a fin de generar un modelo de la misma que será el input al método de Benchmark elegido (simulaciones, métodos analíticos, etc.)



Realizar benchmark

- Someter el elemento a las pruebas
- Medir las respuestas para cada variable analizada y determinar la diferencia respecto a lo que se compara.

Analizar resultados

- Descartar elementos estudiados que en base a los resultados, no cumplen con las necesidades.
- Realizar informes para mostrar los resultados obtenidos.
- Determinar si se requiere recalibrar el benchmark e iniciar un nuevo ciclo de evaluación. Recalibrar el benchmark puede involucrar la corrección de parámetros de prueba, modificación de los lotes de prueba utilizados, etc.
- Desarrollar planes de acción en base a los resultados obtenidos.

La importancia del [benchmarking](#) no se encuentra en la detallada mecánica de la comparación, sino en el impacto que pueden tener estas comparaciones sobre los comportamientos. Se puede considerar como un proceso útil de cara a lograr el impulso necesario para realizar mejoras y cambios.

Dado que la evaluación es de tipo comparativo y decimos que podemos realizar comparaciones de cualquier tipo, entonces deberíamos especificar los tipos de Benchmark existentes.

En lo referido a sistemas, los benchmarks suelen realizarse mediante programas que miden prestaciones del hardware o de parte de él. Algunos indican la velocidad de resolución de operaciones matemáticas, rendimiento en cuanto a la resolución de gráficos, etc.

TIPOS DE BENCHMARK

- **Sintéticos:** están especialmente diseñados para medir el rendimiento de un componente individual del hardware, normalmente llevando el componente escogido a su máxima capacidad. Tienen la intención de medir uno o más características de un sistema, procesador, o compilador. Pueden imitar sets de instrucciones de “aplicaciones reales” o generar sets de instrucciones “artificiales”. Este tipo de benchmark es útil para el debugging o aislamiento de características en especial.

Ejemplos:

- **Whetstone:** Fue el primero de este tipo de Benchmarks. Aún se utiliza ampliamente, porque provee una medida muy razonable de rendimiento de monoprocesadores de aritmética flotante. Uno de sus mayores usos actualmente es como parte de otros benchmarks.

La unidad de medida es *MWIPS: Millones de Instrucciones Whetstone Por Segundo*.

Una instrucción Whetstone puede definirse como una instrucción de punto flotante promedio y se calcula como:

$$\text{MWIPS} = 100 * \text{cant de iteraciones} * \text{cant de WIPS por iteración} / \text{tiempo de ejecución}.$$

- **Dhrystone:** Benchmark sintético para medir la velocidad del sistema en cuanto a rendimiento no numérico. Mide la capacidad del procesador. Es el sucesor de *Whetstone*. La versión original fue realizada en lenguaje ADA aunque actualmente se utiliza en lenguaje C.

La unidad de medida es *DPS: Instrucciones Dhrystones Por Segundo*.

El rendimiento de Dhrystone se calcula a partir de la siguiente fórmula:

$$\text{DPS} = \text{reloj del procesador} * \text{cant de iteraciones} / \text{tiempo de ejecución}.$$

- **Aplicaciones:** herramientas basadas en aplicaciones reales, simulan una carga de trabajo para medir el comportamiento global del equipo. Se emplean en general para la evaluación de sistemas en forma global por lo que tienen gran cantidad de código y de requerimientos de almacenamiento de datos.
Responden a cuestiones como ¿Este equipo X es más rápido que el equipo Y? o ¿Puede mi aplicación ejecutarse más rápidamente en el caso de aumentar la velocidad del procesador, o está limitada por otros subsistemas.
- **Test de Bajo nivel:** Miden directamente el rendimiento de los componentes Ejemplo: el reloj de la CPU, los tiempos de la DRAM y de la caché SRAM, tiempo de acceso medio al disco duro, latencia, tiempo de cambio de pista, etc.
- **Test de Alto nivel:** Están más enfocados a medir el rendimiento de la combinación componente/controlador/SO de un aspecto específico del sistema, como por ejemplo el rendimiento de E/S con archivos, o el rendimiento de una determinada combinación de componentes/controlador/SO/aplicación. Ejemplo: Velocidad de compresión zip

OTROS TIPOS DE BENCHMARK

- Consumo de energía
- Cumplimiento con estándares ambientales, contenido de materiales y manejo del final del ciclo de vida del producto
- Disipación de calor
- Redes
- Reducción de ruido
- Trabajo compartido: mide el rendimiento en las modernas tecnologías de distribución de procesos
- Servidores
- Soporte técnico

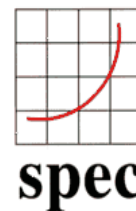
- etc.

BENCHMARKS MÁS CONOCIDOS

- **HINT - Hierarchical INTegration:** A diferencia de los benchmarks tradicionales, HINT no fija el tamaño del problema ni el tiempo de cálculo sino que en su lugar utiliza una medida llamada QUIPS - QQuality Improvement Per Second. Lo que se hace es resolver un problema matemático cuya solución se puede ir mejorando paulatinamente permitiendo explorar los límites computacionales de los computadores así como la calidad del resultado. Se calculan los límites racionales inferior y superior de:

$$\int_0^1 \frac{1-x}{1+x} dx$$

- **Standards Performance Evaluation Corporation (SPEC):** Organización sin fines de lucro que establece y apoya un set de benchmarks estandarizados que pueden ser aplicados a la nueva generación de computadores de alto rendimiento. SPEC desarrolla benchmarks y hace públicos sus resultados. Está compuesto por gran cantidad de empresas: HP, MIPS, Sun, AT&T, Compaq, IBM, Intel, Motorola, DEC, etc.



- **BAPCo - Business Applications Performance Corporation:** Organismo integrado por numerosas empresas de tecnología: AMD, Apple, ARCintuition, Compal, Dell, Hewlett-Packard, Intel, Lenovo, Microsoft, NVIDIA, Samsung, Seagate, Sony, Toshiba, and VIA Technologies, que entre otros benchmarks, desarrolla el llamado **Sysmark**. Este test es de aplicación y permite analizar el rendimiento de programas relacionados con la creación de video, modelado 3D, etc. (Macromedia Dreamweaver, Flash, Adobe Photoshop, Premiere, etc.)



- **3DMark:** Es un soft que permite realizar Benchmark sintético para analizar rendimiento 3D sobre tarjetas gráficas y ordenadores. Hay páginas Web que evalúan la performance de los videojuegos en base a las respuestas obtenidas por este software.
- **Khornerstone:** Es un Benchmark multipropósito que versiona las características de [Dhrystone y de Whetstone](#).
- **Linpack:** Permite medir la eficiencia de sistemas multiprocesadores. Este benchmark describe la performance para resolver un problema de matrices generales $Ax = b$ a tres niveles de tamaño: problemas de 100×100 , problemas de 1.000×1.000 y un problema paralelo escalable. Los resultados son expresados en MFlops (millones de operaciones de coma flotante por segundo).
- **Aquamark:** es un software de Benchmark vuelto para la evaluación de requisitos para ejecución de juegos en ordenadores, evaluando desempeño gráfico y de procesamiento.
- **TPC - Transaction Processing Performance Council:** Es una organización sin fines de lucro que define benchmarks de medición de procesamiento de transacciones en bases de datos. Está compuesto por varios programas.

- TPC-C
- TPC-E
- TPC-H



Se aplican en entornos transaccionales (OLTP: *on-line transaction processing*).

TPC – C es el benchmark de base de datos más reconocido y utilizado en el mercado. Mide los resultados en *transacciones por minuto* (tpmC).

Es la evolución de benchmarks menos complejos y ya obsoletos como son TPC-A, TPC-B, TPC-App, etc.

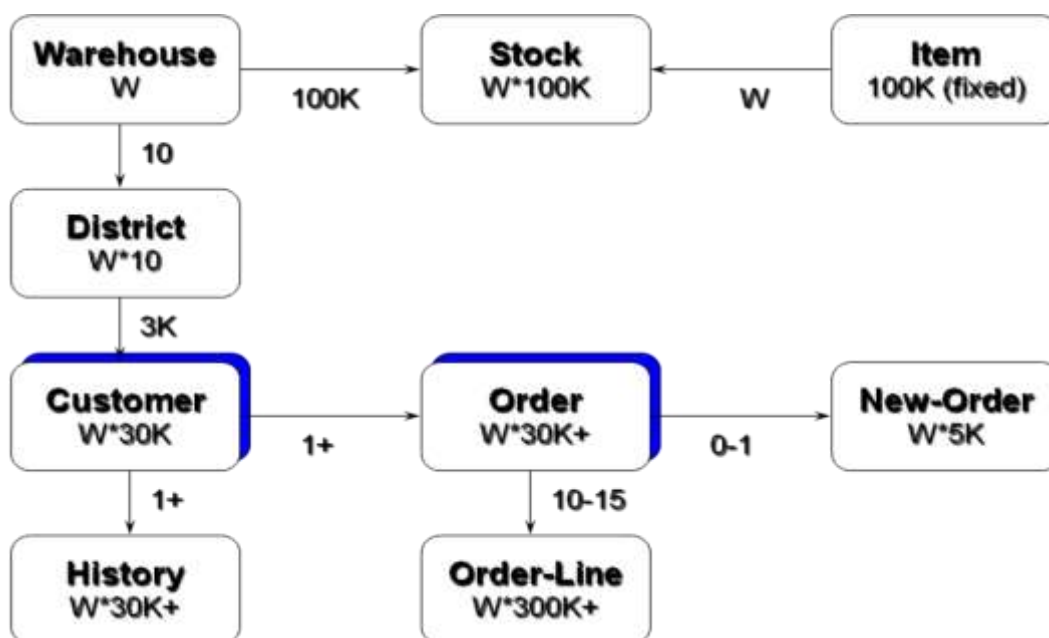
TPC-C simula la actividad de una empresa de venta al por mayor con almacenes y distritos de ventas distribuidos geográficamente en donde los usuarios ejecutan transacciones a la base de datos. Se simula la carga de trabajo (workload) mediante un conjunto de terminales que solicitan la ejecución de una mezcla de cinco tipos de transacciones:

- *New-Order*: Introduce una nueva orden.
- *Payment*: Pago de una orden.
- *Order-Status*: Consulta del estado de una orden.
- *Delivery*: Reparto de las órdenes de un almacén.
- *Stock-Level*: Consulta las existencias de los artículos.

La base de datos está compuesta por 9 tipos de tablas de amplia gama de tamaño y complejidad.

Para el 90% de cada tipo de transacción el tiempo requerido de respuesta es de 5 segundos o menos, excepto el nivel de stock que debe ser menor o igual a 20 segundos.

Esquema de base de datos TPC-C:



TPC-C exige que las transacciones simuladas "sean" ACID:

• A
t
o
m
i
c
i
d
a
d

: Propiedad que verifica que todas las transacciones se confirmen o se aborren. Ante un fallo del sistema no puede quedar ninguna a medias.

- **Consistencia:** Propiedad que verifica que solo se ejecutan aquellas operaciones que no van a romper la reglas y directrices de integridad de la base de datos.
- **Aislamiento (Isolation):** propiedad que asegura que una operación no puede afectar a otras. Esto asegura que la realización de dos transacciones sobre la misma información sean independientes y no generen ningún tipo de error.
- **Durabilidad:** es la propiedad que asegura que una vez realizada la operación, ésta persistirá y no se podrá deshacer aunque falle el sistema. Debe demostrar que puede recuperarse ante:
 - Pérdida de energía
 - Pérdida de memoria

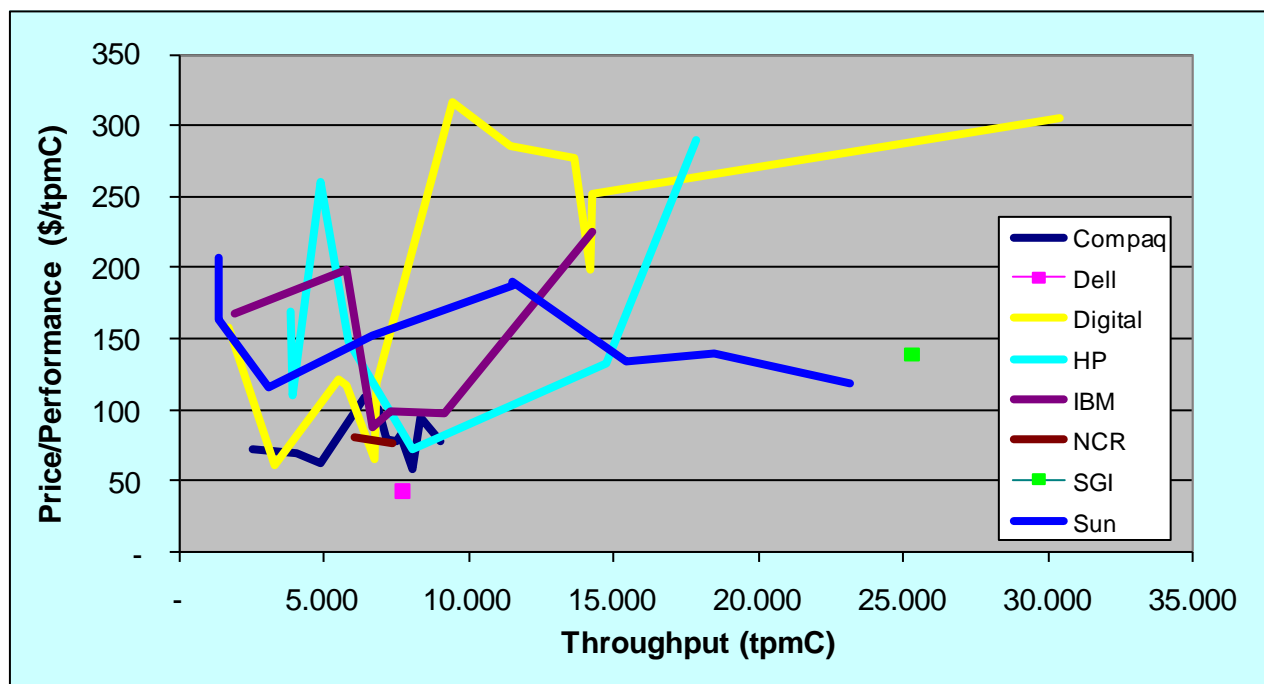
- Pérdida de medios (ej: disco)

Reglas básicas:

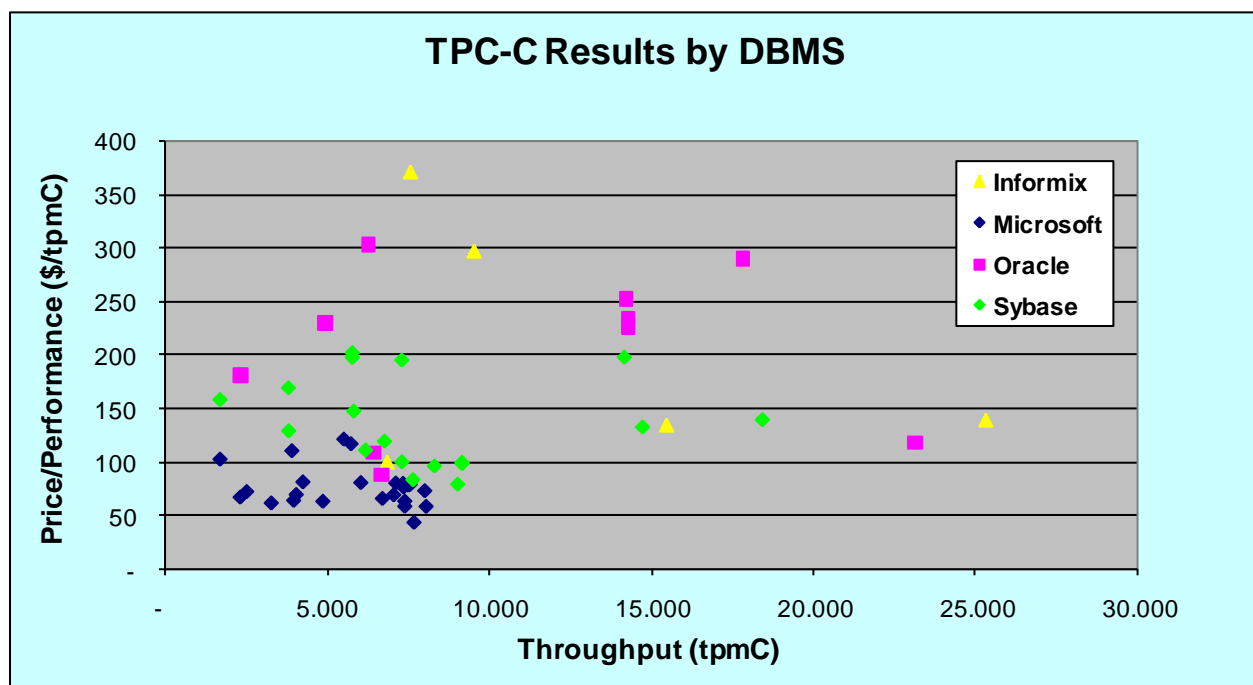
- 1.2 tpmC por Usuario/terminal (máximo)
- 10 terminales por repositorio
- 65-70 MB/tpmC capacidad de disco (mínimo)
- 0.5 IOs/sec/tpmC (típico)
- 250-700 KB memoria principal/tpmC
- Por lo tanto, las reglas básicas para un sistema de 10000 tpmC son:
 - Cuántas terminales? $8340 = 10000 / 1.2$
 - Cuántos repositorios? $834 = 8340 / 10$
 - Cuánta memoria? 2.5 - 7 GB
 - Cuánto disco? $650 \text{ GB} = 10000 * 65$

Resultados comunes:

Por Server:



Por Motor de Base de datos (DBMS):



Sitios Web relacionados:

<http://www.zonebench.com/>

<http://www.tpc.org/>

<http://www.bapco.com/>

<http://www.spec.org/>

<http://hint.byu.edu/>

<http://www.futuremark.com/>