

TEMA 7 SELECCION Y CONFIGURACION DE COMPUTADORES. BENCHMARKING

¿Que es un benchmark? (Comparador de rendimiento)

Un benchmark es un conjunto de procedimientos (programas de computación) para evaluar el rendimiento de un ordenador. Hay cuatro categorías generales de pruebas de comparación :

- Pruebas aplicaciones-base (**application-based**) las ejecuta y las cronometra.
- Pruebas playback (**playback test**), las cuales usan llamadas al sistema durante actividades específicas de una aplicación(Ej.: Llamados a gráficos o uso del disco) y las ejecuta aisladamente.
- Prueba sintética (**synthetic test**) , la cual enlaza actividades de la aplicación en subsistemas específicos.
- Prueba de inspección (**inspection tests**), la cual no intenta imitar la actividad de la aplicación, sino que las ejecuta directamente en los subsistemas específicos.

Los test de aplicaciones base entregan la mejor forma de medir el rendimiento completo de el sistema en el mundo real.

Por ejemplo: El programa **Winstone** de Zdnet, ejecuta mas de una docena de las aplicaciones mas populares en el ambiente Windows, es un ejemplo de este tipo de comparadores.

Donde sea factible la tecnología playback le da la manera mas real de medir subsistemas individuales en aislación.

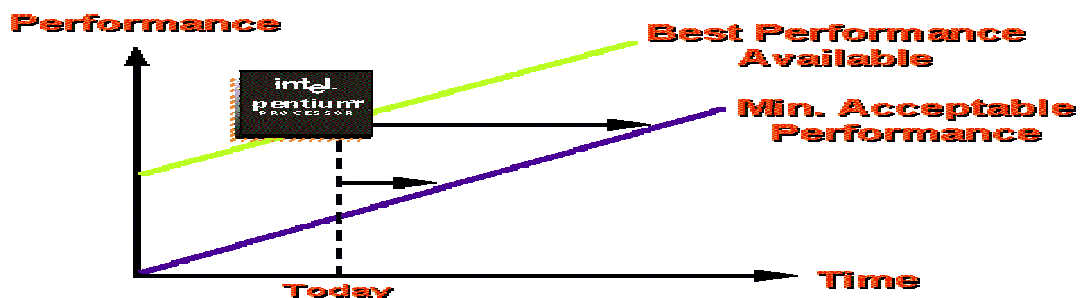
El programa **WinBench** de ZDnet utiliza la tecnología playback para probar gráficos, Cd-rom y subsistemas de disco duro, también corre cientos de otras pruebas en áreas específicas del computador.

Los test Synthetic continúan en el estado de medición del rendimiento es por eso que winbench usa las pruebas de procesadores.

Los test de inspección tienen su lugar verificando el comportamiento libre de fallas y midiendo rendimiento operación por operación, por esto se incluye el test de inspección en el winbench.

Usando el rendimiento para elegir el sistema correcto (¿Porqué considerar el Rendimiento?)

Juzgar el rendimiento de un sistema cuando se están tomando decisiones de compra es algo crítico a fin de retardar la obsolescencia y proteger su inversión.



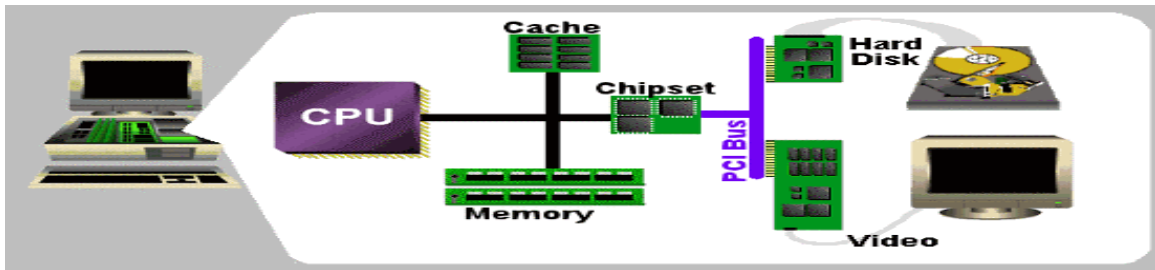
Por ejemplo: Los procesadores Pentium(R) ofrecen la vida útil más larga.

Ud. quiere obtener el máximo en la compra de su sistema. Parte de esto consiste en asegurarse que el sistema que elige le permite hacer lo que desea durante toda la vida del sistema. Esto significa que no solo va a gestionar el software de hoy sino que dispone de los recursos necesarios para ejecutar los mas complejos programas del día de mañana.

Tal como se muestra en el diagrama de más arriba, para cualquier punto determinado en el tiempo se encuentra disponible una gama de procesadores-- aquellos que ofrecen el máximo rendimiento posible y aquellos que ofrecen un mínimo nivel de rendimiento. Las flechas indican que comprando el de mayor rendimiento se obtiene una vida útil más larga.

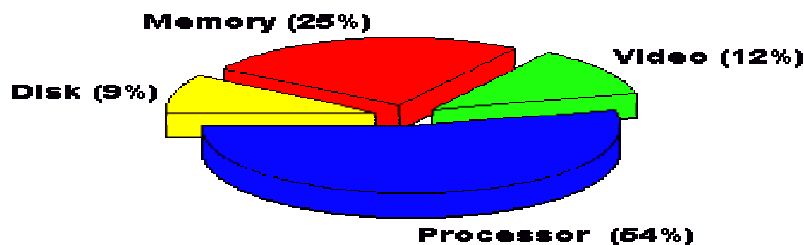
Es importante evaluar a una serie de aspectos relativos al rendimiento cuando se han de tomar decisiones de compra.

Factores del rendimiento de una computadora



El rendimiento global que puede Ud. obtener de su PC depende de cómo cada uno de sus componentes trabajan juntos para realizar una tarea. El efecto de cada componente individual puede variar dependiendo de en cuanto está interviniendo en ejecutar una aplicación particular. Como se muestra más abajo, el 54% del rendimiento del sistema depende de su procesador , o CPU, cuando se ejecutan aplicaciones Windows* típicas. Otros componentes tales como memoria, vídeo y disco también juegan un papel. Una "conexión débil" en el sistema puede reducir significativamente la velocidad y el tiempo de respuesta que Ud. experimente.

Typical Windows* Application Profile



Los porcentajes representan la utilización de los componentes cuando se ejecutan aplicaciones Windows* típicas

Por tanto, cuando elija un sistema, debe de empezar Ud. buscando cual es el procesador que le ofrece el mejor rendimiento para su software, a un precio razonable. Entonces

puede Ud. empezar a analizar los otros componentes del sistema para asegurarse que se benefician de todas las ventajas de la potencia del procesador.

¿ Qué es lo que tiene que tener en cuenta acerca del rendimiento ?

El decidir qué procesador Intel y qué sistema es el adecuado tiende a ser cada vez más difícil dado los nuevos productos PC que están apareciendo en el mercado. Una manera de diferenciar estos productos consiste en saber cómo su rendimiento se ve afectado con los trabajos de los que está Ud. interesado en obtener un mejor rendimiento, a fin de ganar tiempo de respuesta en las actividades que diariamente realiza, tales como crear un documento o trabajar sobre el último programa multimedia. También puede Ud. precisar de un rendimiento que le posibilite la opción de ejecutar el software cada vez más complejo de hoy día. Saber cómo el rendimiento del procesador y el sistema se puede ver afectado, le ayudará a realizar una elección mejor informada. Sin embargo, medir el rendimiento no es siempre algo simple. Para tomar la decisión correcta, necesita Ud. comprender qué es lo que está midiendo.

Un error frecuente es medir el rendimiento entre PCs estableciendo la comparación entre la velocidad del procesador en MHz o velocidad del reloj. Es algo así como utilizar RPM para medir la velocidad de su automóvil.

¿Miraría Vd. el tacómetro de su automóvil, que mide RPM, para determinar si está excediendo el límite de velocidad (y evitándose una elevada multa)? Probablemente no, dado que el tacómetro NO mide la velocidad del vehículo, sino que mide cuan rápido está rotando el motor.

Así como las RPM no indican cuan rápido está circulando su automóvil, los megahertzios no indican lo rápido que su hoja de cálculo se ejecutará. Es, por lo tanto, inadecuado comparar cuan rápido dos PCs realizarán una tarea comparando únicamente velocidades de reloj o megahertzios. Por ejemplo, un procesador Pentium(R) corriendo a 75 megahertzios supera fácilmente a un procesador IntelDX4(TM) a 100 megahertzios. Estas es una de las razones por las que la industria creó las pruebas de rendimiento.

Pruebas comparativas

Una prueba comparativa es un programa software que mide el rendimiento de una computadora, o las partes que la componen, ejecutando exactamente la misma cada vez. Estas pruebas no solamente pueden ayudarle a comprender el rendimiento de los diferentes sistemas a comparar, sino que también pueden ayudarle a evaluar el rendimiento de un sistema determinado bajo diferentes configuraciones hardware y software. Cada prueba evalúa un diferente tipo de trabajo. Algunas pruebas evalúan lo rápido que una computadora puede generar un documento. Algunas pruebas evalúan lo rápido que se dibuja un gráfico en la pantalla. Otras evalúan cómo de rápidas se realizan determinadas operaciones matemáticas . Algunas pruebas pueden evaluar todas esas tareas al mismo. Para obtener resultados que tengan algún significado para Vd., use pruebas que reflejen el uso que hace Vd. de su computadora.

- Hay dos niveles de pruebas: **COMPONENTE y de SISTEMA**.
Las pruebas de componente evalúan solamente partes específicas de una computadora, tales como el procesador, el disco, la tarjeta gráfica, etc.
Las pruebas de sistema evalúan el rendimiento global del sistema.
- Hay dos tipos de pruebas: **APLICACION y SINTETICAS**.

Pruebas de aplicación evalúan el rendimiento usando aplicaciones reales en la forma en que se utilizan.

Las pruebas sintéticas usan programas software creados específicamente para evaluar el rendimiento. Ambos tipos tienen ventajas y desventajas.

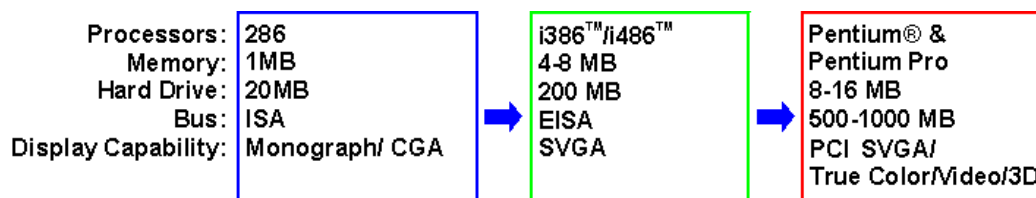
Hay muchas pruebas de rendimiento diferentes disponibles hoy día, pero no todas las pruebas han sido creadas de la misma forma. Crear una prueba requiere una considerable experiencia e inversión. La utilidad y exactitud de cualquier prueba depende fundamentalmente de la calidad de su implementación y las suposiciones que se hacen, por lo que puede darse que los resultados pueden variar de una prueba a otra. Es importante que, como hace normalmente con su software cuando se le queda obsoleto, consiga Vd. versiones actualizadas de las pruebas.

Las pruebas comparativas modernas

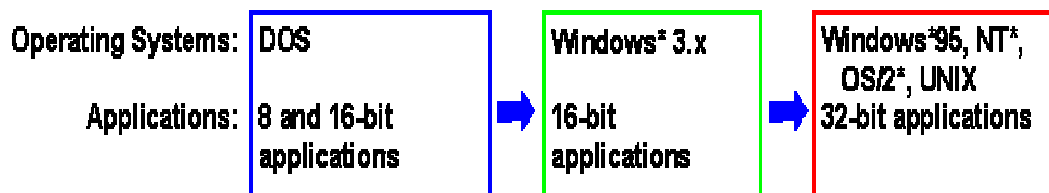
Como medir el rendimiento de los PC de hoy: La tecnología actual suministra rápidos procesadores de 32-bits, mejor memoria, y subsistemas mejorados de display. Estos avances han coincidido con nuevos sistemas operativos, aplicaciones y características, lo que da como resultado una continuada necesidad de pruebas comparativas modernas para medir adecuadamente el rendimiento de los PCs de hoy y mañana.

Cuando se compra un PC, es crítico incorporar a su proceso de evaluación métodos actualizados para obtener el rendimiento. Esto se debe a que pruebas obsoletas basadas en antiguos entornos hardware y software, no pueden decirle todo lo que Vd. necesita saber.

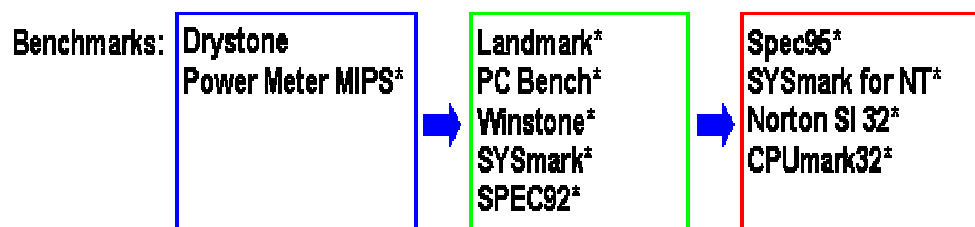
Ante todo, los procesadores y las plataformas son cada vez más sofisticados . . .



Al mismo tiempo también fue evolucionando los sistemas operativos:



El resultado: una continua necesidad de nuevas pruebas comparativas para medir el rendimiento.



Las antiguas pruebas suelen ser inadecuadas

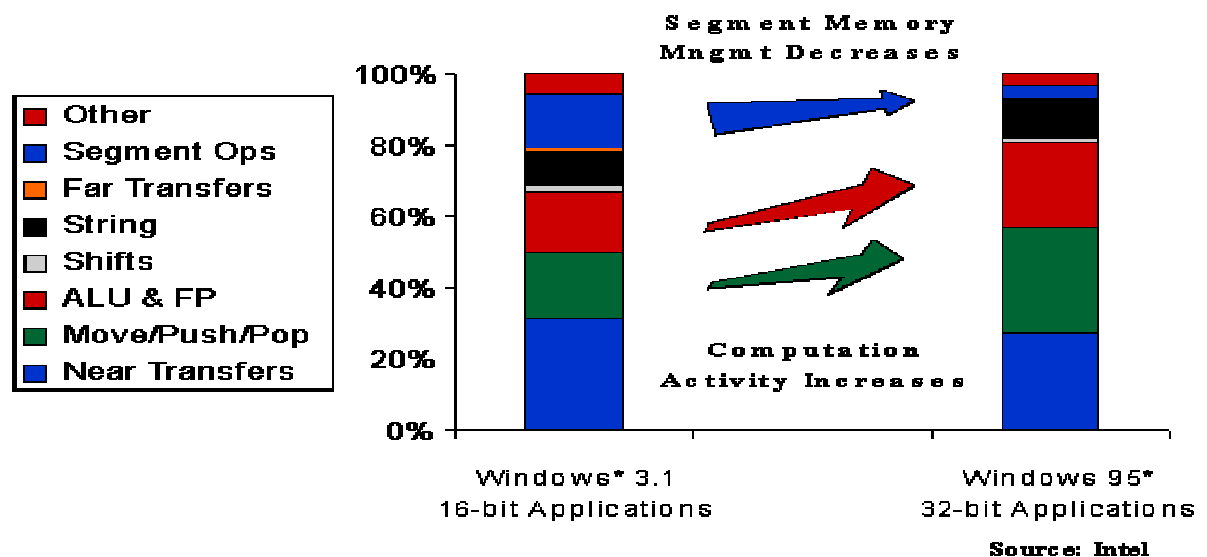
Ese tipo de pruebas no tienen completamente en cuenta la diferencia de rendimiento entre el software de 16 y 32-bits, ni son capaces de testear las posibilidades globales de las plataformas o los sistemas operativos de 32-bits de hoy día. Dos factores contribuyen principalmente al declive de este tipo de pruebas para medir el rendimiento. Ante todo, hay significativas diferencias entre el software para 32-bits y 16-bits. Y los avances en el hardware y los sistemas operativos están creando una gama de nuevas características del software que este tipo de pruebas antiguas sencillamente no tienen en cuenta.

Diferencias entre software de 32-bits y 16-bits

La continua evolución a sistemas operativos de 32-bits está alentando a los vendedores de software a crear nuevas aplicaciones escritas en el modelo de programación para 32-bits. El uso de software de 32-bits tiene implicaciones importantes para el procesador de los PC's dado que el sistema usa el tiempo de proceso de forma diferente --y de forma más eficaz-- cuando ejecuta aplicaciones de 32-bits.

- El software de 16-bits necesita más tiempo en gestionar todo el entorno (tal como la gestión del segmento de memoria, por ejemplo). Esto proviene de los tiempos de programación para el 286 y el DOS, donde solamente era posible un direccionamiento de memoria de 64K al mismo tiempo.

Instruction Mix Comparison



Comparativamente, el software para 32-bits típicamente precisa menos gestión de memoria, liberando al procesador para gastar más tiempo en actividades críticas, tales como tratamiento de instrucciones ALU y coma flotante. El gráfico de más abajo muestra el porcentaje del tiempo que el procesador ha gastado en diferentes instrucciones para aplicaciones de 32-bits en Windows*95, así como para aplicaciones de 16-bits en Windows 3.1.

Esto significa que todas las aplicaciones de 32-bits se ejecutan más rápidamente que todas sus equivalentes de 16-bits, con lo que típicamente las aplicaciones de 32-bits

deberán de tener una significativa mejora sobre la versión para un entorno DOS de 16-bits.

Nuevas características del Software

La mayoría de las pruebas antiguas pueden también fallar al analizar las nuevas características ofrecidas por el software más recientes como:

- **Multitarea**

La mayoría de los sistemas operativos de 32-bits dividen el tiempo de proceso en tareas múltiples --muchas de las cuales pueden ejecutarse de forma continuada en segundo plano-- tales como impresión o conmutar entre distintas aplicaciones. Este tipo de trabajo en modo múltiple requiere una gran potencia de proceso.

- **Aplicaciones más grandes**

El mayor número de características de una aplicación obligan a consumir un mayor espacio en todos los subsistemas de memoria. Esto aumenta los requerimientos de rendimiento para discos, la memoria y la arquitectura de la gestión de memoria.

- **Incremento del tamaño de los datos**

El aumento de aplicaciones que hacen un uso intensivo de vídeo y gráficos, así como velocidades más rápidas de red, requiere operaciones de entrada/salida más eficaces para un ancho de banda más alto del sistema.

- **Multitarea / Multiproceso**

La multitarea tiene una mayor eficacia ejecutando algunas aplicaciones modernas. La evaluación del rendimiento de aplicaciones multitarea y sistemas multiprocesador requiere técnicas especiales.

- **Mejor protección de memoria**

Los nuevos sistemas operativos como Windows NT dotan al sistema de una mayor fiabilidad para aislar las aplicaciones, lo que es por aplicaciones aislantes desde el uno al otro y el sistema operativo -- otra característica que requiere adicional capacidad de proceso.

- **Nuevos recursos del sistema**

Estos Sistemas Operativos que aparecen incluyen nuevos Interfaces de Programación de Aplicaciones (APIs) que hacen más fácil a los desarrolladores de software integrar las más nuevas posibilidades en sus aplicaciones. Estas incluyen mejores gráficos 3D, protocolos de telefonía (TAPI), y soporte de datos más complejos tales como audio y vídeo, que precisan de una mayor potencia de computación.

En resumen, estas pruebas antiguas solamente podrán mostrar el rendimiento del software antiguo. Las pruebas comparativas modernas son necesarias para medir el rendimiento del software moderno.

¿Porqué se necesitan pruebas comparativas modernas?

La gente quiere que sus computadoras duren el mayor tiempo posible. Por esta razón tiene sentido comprar máquinas que tengan el mayor rendimiento disponible. Pero esto precisa de pruebas comparativas diseñadas para las aplicaciones de hoy, a fin de poder evaluar el rendimiento de proceso.

A continuación le indicamos en qué le pueden ayudar las pruebas comparativas modernas a medir el rendimiento de proceso:

- Adecuar apropiados estándares de rendimiento que tomen en cuenta las diferentes características del software de 32-bits y las nuevas posibilidades de multitarea, video y 3D que las viejas pruebas no tienen en cuenta.
- Una medida más precisa de las diferencias de rendimiento relativo entre dos PCs distintos, permitiéndole aumentar la vida útil de un PC con la vista puesta en el futuro.
- Ayudarle a prevenir que tenga que aumentar su inversión antes de tiempo, mostrándole la verdadera diferencia de rendimiento entre dos sistemas.

Las nuevas Pruebas Comparativas que hoy puede Ud. utilizar

Pruebas a nivel de procesador

- SPECint*95, SPECfp*95.
 - Actualización de SPEC 92 del consorcio de la industria SPEC
 - Una prueba CPU/memoria basada en aplicaciones reales
- Norton SI 32*
 - Prueba sintética de 32-bits de Symantec basada en aplicaciones Windows
- CPUMark32*
 - Prueba sintética de 32-bits de Ziff-Davis basada en aplicaciones de 32-bits .
 - Índice iCOMP® 2.0
 - Actualización del actual 'Intel Comparative Microprocessor Performance Index', haciendo énfasis en pruebas de 32-bits

Pruebas a nivel de Sistema

- SYSmark*32 para Windows 95* .
 - Una prueba para aplicaciones Windows 95 •Disponible
- SYSmark para Windows NT*
 - Una prueba de 32-bits a nivel de sistema de BAPCo
 - Aplicaciones reales Windows NT, algunas multitarea
 - Prueba a nivel de sistema solo para 32-bits

Otras áreas del rendimiento del PC necesitan ser exploradas e incorporadas a las pruebas comparativas de la industria para que se pueda disponer de una correcta visión de las capacidades del sistema. Estas áreas incluyen la multitarea, vídeo, gráficos 3D y sonido, comunicaciones y reconocimiento de voz.

Intel está involucrándose activamente, junto con muchos consorcios y organizaciones de la industria, en ayudar a desarrollar estas pruebas. Como resultado, podrá Vd. tener un mejor indicador para estar informado en la decisión de compra de los PCs de hoy y en el futuro.

Pruebas de Rendimiento

El **rendimiento** de una estación de trabajo se mide analizando una serie de componentes físicos que determinan el rendimiento completo del sistema. A la hora de determinar el rendimiento global de un sistema, también hay que evaluar el sistema operativo, los equipos lógicos de red, los compiladores y las librerías gráficas, etc.

Para la evaluación del rendimiento de los sistemas se utilizan las **Pruebas de Rendimiento**, también conocidas como *benchmarks*. Estas son unas medidas que efectúa la industria para comparar factores de rendimiento y relaciones rendimiento/precio de diferentes modelos de estaciones de trabajo. No obstante, estas evaluaciones no son siempre directamente comparables, y en ocasiones ofrecen poca información, porque las configuraciones con las que se realizan las evaluaciones no son expuestas con claridad.

A pesar de lo comentado, existen varias pruebas estándar que ofrecen datos útiles y válidos. Entre los más comúnmente aceptados y conocidos figuran los que se relacionan a continuación, clasificados según el entorno de trabajo en el que son aplicables.

Benchmarks que miden la eficiencia de ejecución de una serie de instrucciones que hacen un uso intensivo de la UCP. Predominantemente miden el rendimiento de la unidad central de proceso. Estas pruebas son propias de entornos de programación y científico-técnicos:

SPECmarks (SPEC CFP92, SPEC rate_fp92, SPEC CINT92, SPEC rate_int92, SPEC SFS)

Benchmark

Se utiliza como sinónimo de carga de test (kernels, carga sintética, carga sintética natural...)

Programas utilizados para cargar el sistema y medir el rendimiento del sistema o de partes de éste.

Benchmarking: comparar sistemas mediante la obtención de medidas.

Aplicaciones muy variadas:

- Comparar sistemas.
- Problemas de sintonización.
- Planificación de capacidad.
- Comparación de compiladores.
- Pasos en el benchmarking

Objetivos de las mediciones.

Selección de los componentes apropiados para cubrir la finalidad.

Comprobar los aspectos del sistema que se va a estudiar y que pueden influir en las prestaciones.

Toda la información anterior, así como la fecha de aplicación debe presentarse en los resultados.

Análisis de los resultados llegando a comprender las razones de los índices obtenidos.

Aspectos que pueden influir de forma no intencionada en los resultados: versiones de las librerías, opciones de los compiladores, hardware específico, ...

Utilización de los benchmarks

Para la adquisición de equipos. El benchmark deberá reflejar las necesidades del comprador.

Sintonización. Se deben conocer muy bien los bench. para conocer las causas de los diferentes resultados.

Planificación de la capacidad. Se va aumentando la carga para conocer la capacidad que queda disponible.

Comparación de compiladores.

Diseño de sistemas informáticos o procesadores. Los bench. se suelen tomar como entrada

para simuladores.

Factores que influyen en el benchmarking

- Tipo y versión del sistema operativo.
- Compilador. Versión y directivas de compilación.
- Lenguajes de programación.
- Librerías. Pueden ir más rápido con menos precisión. IEEE dicta un estándar para las
- librerías.
- Antememoria (memoria caché).
- Verificación de ejecución correcta del bench. (los resultados son los apropiados).

Es importante comprobar que el benchmark realiza alguna tarea y que se ha resultado bien.

Errores comunes

- Representar en la carga de test sólo comportamientos medios. Si se representan valores medios se pueden dar sincronizaciones en los consumos que no son reales.
- Controlar de manera inadecuada el nivel de carga. Modificamos ciertos parámetros esperando que los resultados no cambien.
- Ignorar los efectos de la caché. Las prestaciones de las cachés son muy sensibles al orden en el que se hace las peticiones. En los estudios de caracterización de la carga se suele perder información relativa al orden en que se producen los consumos.
- Ignorar la sobrecarga introducida en la monitorización.
- No validar las medidas. Errores en la medida suelen pasar inadvertidos.
- No asegurar las mismas condiciones iniciales. Por ejemplo, ejecutar dos veces el benchmark.
- No medir las prestaciones del transitorio. Se suelen tomar las medidas una vez el sistema funciona de forma estable. A veces interesa medir el coste de las operaciones de inicio.
- Almacenar muchos datos pero realizar poco análisis. Normalmente la toma de datos y la experimentación ocupan la mayor parte del tiempo del proyecto. Se debe planificar para que el análisis sea el adecuado.

Juegos de Benchmarking

Algunos ejemplos que pueden llevar a que el resultado de un estudio de benchmarking sea erróneo (Falsear intencionadamente los resultados):

- Utilizar configuraciones diferentes para ejecutar la misma carga en dos sistemas.
- Compiladores diseñados para optimizar la carga.
- Utilizar una secuencia de trabajos sincronizada.
- Selección de la carga de forma parcial (sin verificar su representatividad de la carga real).
- Utilizar benchmarks muy pequeños. Caben en la caché y no enseñan problemas con la memoria.
- Traducciones manuales de los benchmarks para mejorar las prestaciones.

Descripción de algunos benchmarks

1.- Benchmark Generales:

A) Linpack. Argonne National Laboratory. Jack Dongarra 1976. Usados para sistemas científicos y de ingeniería. Resuelven sistemas densos de ecuaciones lineales. Muy vectorizable y dependiente de las operaciones en coma flotante y de las librerías de álgebra.

B) BYTE, PCW, EDN. Propuestos por estas publicaciones. Suelen ser pequeños. Las operaciones que realizan suelen ser variadas:

Función de Ackermann: programa recursivo. Sirve para evaluar la facilidad de los lenguajes en las llamadas a subprogramas.

CRIBA (Sieve): calcula números primos. Se usa para comparar procesadores, pc y lenguajes de alto nivel.

Dhrystone: R.P. Weicker Siemens-Nixdorf Information System. No tiene instrucciones en coma flotante ni llamadas al sistema, gran cantidad de instrucciones de manejo de tiras, pocos bucles (tamaño de caché muy importante). Unidades en Dhrystone/Sg.

Whetstone: H.J. Curnow, B.A. Wichmann National Physical Laboratory (G.B.) 1979. Actualmente en Fortran. Compuesto por módulos con diferentes tipos de consumos. El peso de los módulos se puede ajustar fijando los límites de los bucles. Resultados en MegaWhetstones/Sg. Usa muchas operaciones en coma flotante, muy dependiente de librerías matemáticas, pequeño tamaño, pocas variables locales.

Benchmarks de EDN. Conjunto de programas que comprueban la capacidad de las primitivas del sistemas en distintas actividades. Los programas son escritos por cada usuarios, limitándose EDN a sugerir un ejemplo de cómo deberían codificarse estas tareas.

El Benchmark E (benE) se encarga de realizar la búsqueda de una subcadena de caracteres dentro de una cadena de caracteres.

El Benchmark F (benF) sirve para comprobar las habilidades que tienen las primitivas del procesador para el manejo de bits.

El Benchmark H (benH) es un programa que realiza la inserción de un elemento en una lista enlazada.

El Benchmark I (benI) es el algoritmo de quick sort que comprueba las facilidades del procesador en el manejo de pilas y direccionamiento de dispositivos. El programa consiste en ordenar N registros de 16 bytes, cuya clave de ordenación son los caracteres de cada registro tomados desde el tercero hasta el noveno.

El benchmark K (benK) es un algoritmo para la trasposición de matrices de bits. El programa sirve para comprobar las facilidades del procesador en el manejo de bits, así como en el manejo de bucles.

FIBO. Es un programa escrito en C que, de forma recursiva, calcula 100 veces el número de Fibonacci, menor de 24, que es el máximo que se puede alcanzar con 16 bits. Comprueba la facilidad de cálculo del procesador, las llamadas de procedimiento y el manejo de pilas.

FLOATEST. Es un bucle (programa) en el que se realizan multiplicaciones y divisiones en coma flotante. Sirve para ver la capacidad de cálculo del procesador en coma flotante.

GRAFSCRN. Es un programa para dibujar en pantalla con la función $\text{dot}(x,y)$.

Hilbert y Sierp. Son dos programas de dibujo para comprobar las habilidades gráficas de los sistemas, así como su capacidad de cálculo y las llamadas a procedimientos.

INTMATH. Este programa repite 1000 veces la operación $X + (Y * Y - Y) / Y$ usando datos de tipo entero. Está escrito en C y en Fortran, y sirve para comprobar la capacidad de cálculo entero del procesador.

REALMATH. Este programa es exactamente igual que el Intmath, pero usando datos de tipo flotante.

SAVAGE. Es un programa de cálculo consistente en un bucle que repite 1000 veces la operación $\tan/\text{atang}(\exp(\log(\sqrt{a+a}))) + 1.0$, de forma que es un procedimiento útil para comprobar el uso de librerías de cálculo.

STORE. Es un programa para comprobar la capacidad de acceso a disco que tiene el sistema. El programa crea un fichero, y escribe 1000 veces la tira *1234567890qwertyuiop*. El programa está escrito en C.

TEXTSCRN. Es un programa similar al store, pero la escritura de la información se realiza en la pantalla, por lo que es un programa útil para comprobar el sistema de vídeo del equipo. Está escrito en C y FORTRAN.

C) SSBA1.21E: paquete para la evaluación de sistemas UNIX. Diseñado por la asociación

francesa de usuarios de UNIX. Compuesto por 12 programas. Testean los siguientes aspectos:

- Potencia CPU.
- Implementación del sistema en general y del sistema de ficheros.
- Compiladores de C y Fortran.
- Manejo de memoria y rendimiento de las cachés.
- E/S de disco y rendimiento del controlador.
- Rendimiento multiusuario frente a tareas significativas.

BSD. Fue diseñado por Karels, McUsick y Leffler para medir las mejoras de rendimiento entre versiones del sistema operativo BSD 4.2 y BSD 4.3.

BYTE. Está enfocado hacia microp procesadores. Aquí no se compila nada, sino que se ejecuta un script con ordenes INUX para el manejo de tiras de caracteres. Primero con una tarea, luego como 2,4,8,16 tareas concurrentes para medir el factor de carga. Este test da información sobre las capacidades multiusuario, así como sobre la calidad de la implantación de UNÍS y de la velocidad de la CPU.

MIPS. Diseñado por Bill Joy, consiste en calcular los primeros 99 decimales de la raíz cuadrada de 2.

DODUC. Proviene de un programa de simulación usado ampliamente en ciencias energéticas (programa Montecarlo). Este programa es muy sensible al tamaño de la cache y al uso de coprocesador matemático. Es un programa útil para probar el

compilador Fortran; además se debe ejecutar con una precisión de 64 bits y preferentemente con optimización.

MUSBUS. Diseñado por Ken McDonell en la Universidad Monash en Australia. Testea la productividad de discos y la simulación de multiusuario. Se compila, se crean los directorios y la carga de los usuarios simulados y se ejecuta la simulación tres veces, midiendo la CPU y el tiempo transcurrido.

SAXER. Presentado por John Saxer en la Conferencia de Usenix de Dallas (1984), este test mide la productividad de disco. Compila ocho ficheros fuente y después se ejecutan alternativamente para frustrar las ventajas de la cache. Este test mide el rendimiento combinado del hardware (el disco y su controlador) y del software (el sistema de ficheros). También puede dar indicaciones para estaciones sin disco productividad NSF (Network System File). El valor de productividad se da después de realizar diferentes manipulaciones (crear, escribir, leer, acceso aleatorio, tubería) y la escritura de un fichero de 10 Mbyte.

TESTC. Se ha extraído de aplicaciones GKS reales escritas en C. Consiste en compilar los ficheros fuente con la opción -c (para obtener ficheros .o).

El test sirve para medir el rendimiento del compilador de C, la velocidad del procesador, y el rendimiento del sistema de ficheros. Es sensible al rendimiento de las E/S de disco y del manejo de memoria y al tamaño de los ficheros .o. Este test es representativo de entornos con ficheros fuente grandes.

TOOLS. Comprueba el rendimiento de algunas ordenes básicas y usuales del sistema operativo UNIS, y por tanto, la calidad del UNIS y la velocidad del procesador.

UTAH. Compuesto por una colección de programas realizada por la universidad de Utah. Hay diferentes programas, como por ejemplo el bct.sh calcula la serie de Fibonacci.

2.- Benchmark como programas de aplicación

QNAP2. Es un programa muy usado en modelización. Comprueba el rendimiento de ordenadores en cálculo científico, ya que, además de realizar un cálculo intensivo, también utiliza acceso al sistema de disco. La ejecución de este paquete supone provocar una carga elevada en el sistema.

SMPL. Es útil para la simulación de eventos discretos y de redes de colas. Normalmente lo utilizan diseñadores de sistemas de computadores y comunicaciones para analizar el rendimiento de sus diseños mediante simulación. Básicamente genera números aleatorios. Es de cálculo intensivo, siendo muy sensible a la utilización del coprocesador matemático.

MCNP. (Monte Carlo Neutron). Simula el funcionamiento de un reactor nuclear. Está escrito en Fortran. El acceso a los datos lo realiza solo al principio de la ejecución.

GCC. Es un compilador de C, GNU C, y consiste en compilar una serie de programas a lenguaje ensamblador. El propio GCC está escrito en C y su principal inconveniente es la portabilidad, ya que depende mucho del sistema en que está instalado y se debe disponer de una serie de ficheros de configuración para cada sistema en que se quiera instalar. Sirve para probar la facilidad del sistema en tareas de compilación, y es uno de los benchmarks más usados en los últimos tiempos.

TEX. Es uno de los procesadores de textos más usados.

SPICE. Es un programa de simulación de propósito general, pero con especial énfasis en el diseño de circuitos analógicos. Realiza cálculos enteros y en coma flotante en doble precisión. Como es bastante largo, también sirve para comprobar el rendimiento de la cache.

3.- Benchmark para computadores personales

SPEC (Systems Performance Evaluation Cooperative). Hecho por un grupo de marcas comerciales, intenta ser representativo de la carga actual. Compuesto por 10 programas:

- gcc (compilador).
- Espresso (herramienta de automatización de diseño electrónico).
- Spice 2g6 (herramienta EDA).
- Docuc (programa extraído de un simulador Monte Carlo de reactores nucleares).
- NASA7 (operaciones matriciales en coma flotante).
- LI (programa de las "ocho reinas" usando LISP).
- Eqntott (representar ecuaciones lógicas en tablas de verdad).
- Matrix300 (operaciones con matrices usando rutinas de Linpack en coma flotante) muy vectorizable.
- Fpppp (benchmark de química) poco vectorizable.
- Tomcatv (programa de generación de red vectorizada) muy vectorizable.

TPC. (Transactions Processing Performance Council). Esta orientado a la evaluación del rendimiento y comparación de sistemas de proceso transaccional y base de datos.

Esta compuesto por cinco benchmarks: TPC-A; TPC-B; TPC-C; TPC-D; TPC-E y el TPC-CS

The perfect benchmark club. Consiste en actualizar la carga para realizar pruebas de evaluación del rendimiento, de forma que los tests de carga sean similares a las cargas reales en los sistemas actuales de carácter científico. Básicamente estandariza una serie de programas. Compuesto por trece programas:

ADM: Código tridimensional suministrado por IBM que sirve para simular concentraciones y depósitos de polución mediante resolución de sistemas completos de ecuaciones hidrodinámicas;

ARC3D: Analiza problemas de flujo de fluidos en tres dimensiones resolviendo ecuaciones de Euler y Navier;

BDNA: Realiza simulaciones de dinámica molecular en agua;

DYFESM: Análisis de estructuras simétricas anisotrópicas a partir de elementos finitos dinámicos bidimensionales;

FLO52Q: Analiza el flujo que pasa bajo un airfoil por resolución de ecuaciones inestables de Euler;

MDG: cálculo de dinámica molecular;

MG3D: mide la migración sísmica y se usa para investigar la estructura geológica de la tierra;

OCEAN: resuelve ecuaciones dinámicas bidimensionales de una capa de fluido.

QCD: sirve para observar relaciones entre componentes de la materia molecular;

SPEC77: simula flujo atmosférico;

SPICE: ya descrito anteriormente;

TRACK: sirve para determinar el curso de un conjunto de un número desconocido de objetivos. Ejemplo: propulsores de un cohete.

TRFD: simula aspectos computacionales de la transformación integral de dos electrones.

Formas de aplicar los benchmarks

- Uno detrás de otro. Para ver cuanto tarda un ordenador en ejecutar una colección de
- programas.

- Como los ladrillo en un muro. Para comprobar el comportamiento en multitarea.
- Lanzando peticiones desde otro equipo. Comprobar el funcionamiento en carga interactiva.

Resumen de cosas a comprobar

- Optimizaciones del compilador.
- Comprobar la talla de memoria para estar seguros de lo que se mide.
- Vigilar otros procesos que se están ejecutando.
- Comprobar el tiempo que tarda en ejecutarse y el tiempo que realmente está consumiendo recursos.
- Asegurarse que los resultados son los esperados y con la misma precisión.
- Analizar los resultados en el ordenador actual.
- Se pueden descubrir mejoras en el propio ordenador y dará una idea de las mejoras que se pueden obtener.

Unidades utilizadas en benchmarking

MIPS: Millones de instrucciones por segundo. El ordenador que se toma como referencia es el DEC VAX 11/780 (1978). A veces se le llama VAX MIPS.

- VAX sacaba 470000 instrucciones /sg y apuradito.

Por convenio $1657 \text{ dhrystones/sg} = 1 \text{ MIPS}$.

MFLOPS: Millón de operaciones coma flotante /sg. Son operaciones realizables por hardware. Suelen medirse con programas que son mezclas de sumas y multiplicaciones. En algunas ocasiones se habla de picos de megaflops mayores que la frecuencia del procesador. En cambio ojo con estos valores pico ya que se obtienen bajo cargas muy apropiadas a las características.

MWIPS: Mediadas propias de benchamarck como el Whetstone y el Dhrystone (Mega Whestone instrucciones por segundo) o los DIPS

Creando benchmarks

- Tiempo de ejecución.
- Memoria.
- Utilización de kernels.
- Forma de aplicar los benchmarks.
- Portabilidad.