Números de Latencia

Puntos Clave

- Parece probable que el video discuta los números estándar de latencia que los programadores deben conocer, basándose en su título y contenido en línea relacionado.
- La investigación sugiere que estos números incluyen tiempos para operaciones como el acceso a la caché L1 (0.5 ns) y viajes de red de ida y vuelta (hasta 150 ms), variando según el hardware.
- La evidencia sugiere que estos números son aproximados, con actualizaciones que reflejan avances tecnológicos, especialmente en SSDs y redes.

Introducción

El video “Latency Numbers Programmer Should Know: Crash Course System Design #1”probablemente aborda figuras esenciales de latencia para operaciones informáticas, cruciales para el diseño de sistemas. Estos números ayudan a los programadores a entender los impactos en el rendimiento y optimizar los sistemas.

Números de Latencia y su Importancia

La latencia es el retraso entre iniciar y completar una operación, como acceder a la memoria o enviar datos a través de una red. El video probablemente enumera latencias típicas, como: - Referencia a la caché L1 en 0.5 nanosegundos (ns), el acceso a la memoria más rápido. - Un viaje de ida y vuelta dentro del mismo centro de datos en 500 microsegundos (us) o 0.5 milisegundos (ms), afectando a los sistemas distribuidos.

Estas cifras, aunque aproximadas, guían decisiones en el diseño de sistemas, como elegir entre memoria y almacenamiento en disco.

Contexto en el Diseño de Sistemas

Comprender estas latencias ayuda a optimizar el código, hacer compromisos y mejorar la experiencia del usuario. Por ejemplo, saber que una búsqueda de disco toma 10 ms puede influir en el diseño de la base de datos para minimizar tales operaciones.

Detalle Inesperado

Un aspecto interesante es cómo estos números, como los tiempos de lectura de SSD, han mejorado con la tecnología, mientras que las latencias centrales de la CPU, como el acceso a la caché L1, permanecen estables, mostrando el impacto desigual de la evolución del hardware.

Nota de Encuesta: Análisis Detallado de los Números de Latencia del Video

Esta nota proporciona una exploración exhaustiva de los números de latencia probablemente discutidos en el video "Latency Numbers Programmer Should Know: Crash Course System Design #1", basada en el contenido en línea disponible y los recursos relacionados. El análisis tiene como objetivo sintetizar la información para programadores y diseñadores de sistemas, ofreciendo tanto un resumen como conocimientos detallados sobre la importancia de estos números.

Antecedentes y Contexto El video, accesible en YouTube, es parte de una serie sobre diseño de sistemas, centrándose en números de latencia críticos para los programadores. La latencia, definida como el retraso de tiempo entre la iniciación y la finalización de una operación, es crucial para entender el rendimiento del sistema. Dado el título del video y las búsquedas relacionadas, parece cubrir figuras estándar de latencia popularizadas por figuras como Jeff Dean de Google, a menudo referenciadas en comunidades de programación.

Las búsquedas en línea revelaron varios recursos que discuten estos números, incluyendo un GitHub Gist titulado "Latency Numbers Every Programmer Should Know"(GitHub Gist) y un artículo de Medium de 2023 (Artículo de Medium). Estas fuentes, junto con una publicación de High Scalability de 2013 (High Scalability), proporcionaron una base para compilar el contenido probable del video.

Compilación de Números de Latencia Basado en la información recopilada, la siguiente tabla resume los números estándar de latencia, probablemente discutidos en el video, con explicaciones para cada operación:

Operación	Latencia (ns)	Latencia (us)	Latencia (ms)	Explicación
Referencia a la caché L1	0.5	-	-	Acceso a datos en la caché de nivel 1, la memoria más rápida cerca de la CPU.
Predicción de rama errónea	5	-	-	Penalización cuando la CPU predice incorrectamente una rama condicional.
Referencia a la caché L2	7	-	-	Acceso a datos en la caché de nivel 2, más grande que L1 pero más lenta.
Bloqueo/desbloqueo de mutex	25	-	-	Tiempo para adquirir y liberar un mutex en programas multihilo.
Referencia a la memoria principal	100	-	-	Acceso a datos desde la memoria principal de acceso aleatorio (RAM).
Comprimir 1 KB con Zippy	10,000	10	-	Tiempo para comprimir 1 kilobyte usando el algoritmo Zippy.
Enviar 1 KB de bytes sobre una red de 1 Gbps	10,000	10	-	Tiempo para transmitir 1 kilobyte sobre una red de 1 Gigabit por segundo.

Operación	Latencia (ns)	Latencia (us)	Latencia (ms)	Explicación
Leer 4 KB aleatoriamente desde SSD	150,000	150	-	Lectura aleatoria de 4 kilobytes desde un disco de estado sólido.
Leer 1 MB secuencialmente desde la memoria	250,000	250	-	Lectura secuencial de 1 megabyte desde la memoria principal.
Viaje de ida y vuelta dentro del mismo centro de datos	500,000	500	0.5	Tiempo de viaje de red de ida y vuelta dentro del mismo centro de datos.
Leer 1 MB secuencialmente desde SSD	1,000,000	1,000	1	Lectura secuencial de 1 megabyte desde un SSD.
Búsqueda de HDD	10,000,000	10,000	10	Tiempo para que un disco duro busque una nueva posición.
Leer 1 MB secuencialmente desde disco	20,000,000	20,000	20	Lectura secuencial de 1 megabyte desde un HDD.
Enviar paquete CA->Países Bajos->CA	150,000,000	150,000	150	Tiempo de viaje de ida y vuelta para un paquete de red desde California a Países Bajos.

Estos números, principalmente de 2012 con algunas actualizaciones, reflejan el rendimiento típico del hardware, con variaciones notadas en discusiones recientes, especialmente para SSDs y redes debido a avances tecnológicos.

Análisis e Implicaciones Los números de latencia no son fijos y pueden variar según el hardware y configuraciones específicas. Por ejemplo, una publicación de blog de 2020 de Ivan Pesin (Pesin Space) notó que las latencias de disco y red han mejorado gracias a mejores SSDs (NVMe) y redes más rápidas (10/100Gb), pero las latencias centrales de la CPU como el acceso a la caché L1 permanecen estables. Esta evolución desigual destaca la importancia del contexto en el diseño de sistemas.

En la práctica, estos números guían varios aspectos: - **Optimización del Rendimiento**: Minimizar operaciones con alta latencia, como búsquedas de disco (10 ms), puede mejorar significativamente la velocidad de la aplicación. Por ejemplo, almacenar en caché datos frecuentemente accedidos en memoria (250 us para lectura de 1 MB) en lugar de disco puede reducir los tiempos de espera. - **Decisiones de Compromiso**: Los diseñadores de sistemas a menudo enfrentan elecciones, como usar caches en memoria frente a bases de datos. Saber que una referencia a la memoria principal (100 ns) es 200 veces más rápida que una referencia a la caché L1 (0.5 ns) puede informar tales decisiones. - **Experiencia del Usuario**: En aplicaciones web, las latencias de red, como un viaje de ida y vuelta en el centro de datos (500 us), pueden afectar los tiempos de carga de la página, impactando la satisfacción del usuario. Una publicación de blog de Vercel de 2024 (Blog de Vercel) enfatizó esto para el desarrollo frontal, notando cómo las cascadas de red pueden acumular latencia.

Contexto Histórico y Actualizaciones Los números originales, atribuidos a Jeff Dean y popularizados por Peter Norvig, datan de alrededor de 2010, con actualizaciones por investigadores como Colin Scott (Interactive Latencies). Una publicación de Medium de 2019 de Dan Hon (Medium de Dan Hon) añadió latencias humorísticas pero relevantes, como reiniciar un MacBook Pro (90 segundos), ilustrando retrasos más amplios relacionados con la tecnología. Sin embargo, los números de latencia centrales han visto cambios mínimos, con el GitHub Gist sugiriendo que permanecen “bastante similares”hasta 2023, basados en limitaciones físicas.

Conclusión y Recomendaciones Para programadores y diseñadores de sistemas, memorizar estos números de latencia proporciona un modelo mental para la afinación del rendimiento. Deben tratarse como directrices, con benchmarks reales realizados para hardware específico. Mantenerse al día con actualizaciones, especialmente en tecnologías emergentes como la computación cuántica o redes 5G, será crucial. Recursos como el GitHub Gist y el artículo de Medium ofrecen puntos de partida para una exploración más profunda.

Este análisis, basado en el contenido probable del video y complementado con una investigación en línea extensa, subraya la relevancia perdurable de los números de latencia en la informática, con un llamado a adaptarse a los cambios tecnológicos para un diseño de sistemas óptimo.

Citaciones Clave

- Latency Numbers Every Programmer Should Know GitHub Gist
- Latency Numbers Programmer Should Know Video de YouTube
- Artículo de Medium de Números de Latencia Actualizados
- Más Números que Cada Programador Asombroso Debe Saber High Scalability
- Números de Latencia que Cada Desarrollador Web Debe Saber Blog de Vercel
- Números de Latencia que Cada Ingeniero Debe Saber Blog de Pesin Space
- Más Números de Latencia que Cada Programador Debe Saber Medium de Dan Hon
- Números que Cada Programador Debe Saber por Año Interactive Latencies