Engineer

Latencia

Espera entre la petición y el inicio de la respuesta

Qué es

La latencia es el tiempo que tarda un sistema en empezar a responder: desde que una petición sale del cliente hasta que vuelve el primer byte de la respuesta. En ingeniería también se habla de retardo o retraso, o de tiempo de ida y vuelta (RTT) para una ruta de red medida en ambos sentidos; en rendimiento web, la misma idea se llama tiempo hasta el primer byte (TTFB). El tiempo de respuesta es su hermano mayor: llega hasta el último byte de la respuesta.

Es fácil confundirla con la tasa de procesamiento y el ancho de banda, pero las tres responden a preguntas distintas. El ancho de banda indica cuántos bits puede transportar un enlace por segundo. La tasa de procesamiento indica cuánto trabajo completa un sistema por segundo. La latencia indica cuánto espera una tarea. Un proceso por lotes nocturno puede tener una tasa de procesamiento excelente y una latencia pésima, y un servidor de chat, lo contrario.

Cada milisegundo procede de uno de estos componentes:

- **Propagación**, la señal que recorre la distancia: la luz en fibra óptica recorre aproximadamente 200 km por milisegundo, así que una ruta de 1.000 km cuesta al menos 10 ms de ida y vuelta antes de ejecutar ningún software.
- **Transmisión**, colocar los bytes en el enlace, algo importante para cargas útiles grandes en conexiones lentas.
- **Espera en cola**, detrás de otro trabajo en el búfer de un router, un grupo de hilos, un grupo de conexiones o un bloqueo; es el componente que crece con la carga.
- **Procesamiento**, el trabajo real: la consulta, la plantilla, el cifrado.

- **Viajes de ida y vuelta del protocolo**, la conversación previa al primer byte útil. Una conexión HTTPS nueva sobre TCP necesita primero una negociación TCP y otra TLS, y TLS 1.3 redujo su negociación completa a un único viaje de ida y vuelta.

El grupo de métricas de rendimiento IP de IETF concretó la versión de red en RFC 2681: el retardo de ida y vuelta va desde el primer bit de un paquete enviado hasta el último bit de la respuesta recibida. Dos de sus decisiones se trasladan a cualquier servicio. Una respuesta que nunca llega se cuenta como indefinida, informalmente infinita, en vez de excluirla sin avisar de la muestra. Además, las estadísticas que describe son los percentiles, la mediana y el mínimo. También observa que el mínimo refleja solo la propagación y la transmisión, mientras que todo lo que lo supera indica congestión, una distinción que vale tanto para una API como para una ruta entre routers.

Para una petición web individual, curl divide la espera en sus etapas:

```
curl -s -o /dev/null "$URL" -w '
dns        %{time_namelookup}
connect    %{time_connect}
tls        %{time_appconnect}
first byte %{time_starttransfer}
total      %{time_total}
'
```

Cada cifra expresa los segundos transcurridos desde el inicio, así que las diferencias entre ellas son los costes: resolver el nombre, abrir la conexión, negociar TLS y, después, el tiempo que el propio servidor dedica a trabajar hasta el primer byte.

Utilice el concepto en una decisión

Aplique «¿Qué es la latencia?» a una decisión real. Podemos ayudarle a relacionar el concepto con el sistema que está construyendo o comprando. Monitorización y alertas que avisan cuando un fallo aún es pequeño, y cifras que muestran con qué frecuencia se mantiene así.

Explore nuestros servicios Póngase en contacto →

Compruebe los supuestos detrás del término

Utilice la explicación y los trabajos enlazados para formular preguntas sobre su proyecto. Podemos examinar los supuestos con usted antes de elegir una herramienta. Primer panel de observabilidad de la empresa — la primera visión de la empresa sobre sus propios sistemas

Primer panel de observabilidad de la empresa Póngase en contacto →

Por qué importa

La latencia es lo que percibe el usuario; la tasa de procesamiento es lo que refleja la factura del servidor. Cuando una página es lenta, hay que decidir dónde gastar. Los cinco componentes anteriores necesitan soluciones distintas; cuatro de ellos muestran por qué comprar hardware no es un remedio general. La distancia se resuelve acercándose: una caché, una CDN o una región próxima a los usuarios. Los viajes de ida y vuelta se resuelven reduciéndolos: reutilizar conexiones, agrupar consultas u obtener datos en paralelo. El procesamiento se resuelve haciendo menos trabajo: un índice o un plan de consulta mejor. Añadir capacidad puede reducir la espera en cola; procesadores más rápidos, más memoria o mejores operaciones de entrada y salida también pueden reducir el tiempo de procesamiento cuando esos recursos son el cuello de botella. Un servidor de mayor capacidad no es automáticamente más rápido: mida dónde se consume el tiempo antes de elegir hardware.

Las medias hacen que el problema parezca menor de lo que es, porque los usuarios se encuentran con la cola de la distribución. Una página que llama en paralelo a diez servicios de backend espera al más lento. Si cada llamada tiene, de forma independiente, una probabilidad del 1 % de ser lenta, al menos una llamada lenta afecta a $1 - 0{,}99^{10}$ de las cargas de página, aproximadamente el 9,6 %. Un retraso poco frecuente en las partes se convierte en uno frecuente en el conjunto.

Por eso los objetivos rigurosos se expresan en percentiles, por ejemplo, el 99 % de las peticiones respondidas en un máximo de 300 ms en un período de 30 días, y por eso el percentil 99 suele ser la cifra que conviene vigilar. La mediana indica cómo se percibe el servicio en un buen día; los percentiles altos, cómo lo perciben los clientes que están a punto de marcharse.

Cómo se aplica en la práctica

En una aplicación de mapas geográficos, la latencia al representar capas y consultar datos de ubicación perjudicaba tanto a los usuarios como a la fiabilidad del backend. El trabajo se centró en el procesamiento de consultas espaciales de un backend que utilizaba una base de datos heredada basada en Microsoft SQL Server. El traslado a PostgreSQL con PostGIS redujo el tiempo medio de respuesta de 2,5 segundos a menos de 250 milisegundos, una mejora de 10 veces en el rendimiento de las consultas espaciales. El trabajo incluyó índices

GiST y SP-GiST en las columnas espaciales, la migración de más de 50 millones de registros y ajustes de configuración para el acceso concurrente; la carga de CPU del backend descendió un 65 %.

La latencia también limita la tasa de procesamiento cuando hay competencia por una fila. Una nota sobre vender la última entrada exactamente una vez sitúa el límite de un artículo individual aproximadamente en el número de ventas que permite un bloqueo dividido por el tiempo que se mantiene ese bloqueo. Cualquier operación lenta dentro del bloqueo reduce ese límite: una llamada de pago realizada mientras se mantiene lo deja en unas dos ventas por segundo. Incluso una medida de seguridad tiene un coste aquí. Una réplica de reserva síncrona en PostgreSQL añade latencia a cada confirmación, lo que prolonga precisamente el tiempo de retención del bloqueo por el que se divide para obtener ese límite.

Para averiguar en qué se emplea el tiempo, la monitorización propia de una plataforma registra la latencia de la API en su propia base de datos y traza las peticiones mediante OpenTelemetry, de modo que una petición lenta se puede seguir por el frontend, la API y la base de datos en lugar de adivinar dónde se detiene.

Dónde surgen los problemas

La media oculta la cola de la distribución. Una media de 100 ms es compatible con que una petición de cada cincuenta tarde tres segundos. Registre la latencia en un histograma y consulte sus percentiles. Después, vigile cómo se combinan: no se pueden promediar los percentiles 99 de cinco servidores para obtener el percentil 99 del conjunto. Prometheus lo explica en su documentación: los cuantiles calculados por un resumen no se pueden agregar, mientras que los recuentos de los intervalos de un histograma pueden sumarse primero y calcularse el cuantil después.

El cronómetro empieza en el sitio equivocado. Un servidor que mide desde el momento en que comienza su manejador omite el tiempo pasado en el equilibrador de carga, en la cola de aceptación o esperando una conexión del grupo. Los generadores de carga que esperan cada respuesta antes de enviar la siguiente petición mandan menos peticiones justo cuando el sistema se atasca, así que el atasco apenas aparece en los resultados. Mida también desde el cliente y deje que el servidor indique su parte en una cabecera estándar:

```
Server-Timing: db;dur=53, app;dur=47.2
```

La conversación requiere demasiados intercambios. Una página que ejecuta una consulta por fila de una lista paga un viaje de ida y vuelta por fila. Con la base de datos en otra región y 30 ms por viaje, cien filas cuestan tres segundos antes de que ninguna consulta haga trabajo real. Una consulta con una unión o una llamada por lotes paga la distancia una sola vez.

Los reintentos multiplican la carga. Un cliente que abandona tras un segundo y reintenta tres veces genera cuatro veces el tráfico contra ese servidor lento, lo que lo ralentiza aún más. Dé a cada petición un plazo total, aumente exponencialmente la espera entre intentos con una variación aleatoria y limite el número de reintentos.

Lecturas adicionales

- RFC 2681: métrica de retardo de ida y vuelta para IPPM — el retardo de ida y vuelta definido hasta el bit, con tratamiento explícito de las respuestas perdidas y los percentiles.
- RFC 7679: métrica de retardo unidireccional para las métricas de rendimiento IP — la contrapartida de un solo sentido, publicada como estándar de Internet.
- Server Timing — la cabecera del W3C que permite al servidor indicar al navegador en qué se empleó el tiempo.
- Histogramas y resúmenes — por qué no se pueden combinar los cuantiles de un resumen y sí los recuentos de los intervalos de un histograma.
- RFC 8446: TLS 1.3 — el diseño de la negociación, incluido el viaje de ida y vuelta que elimina.



Glosario

Elija según sus restricciones reales

Describa la ruta de despliegue, los riesgos operativos y quiénes se encargan de las guardias. Podemos definir el alcance de un cambio con comprobaciones, una vía de recuperación y responsabilidades claras.

Vea cuánto cuesta Póngase en contacto →

Haga práctica la terminología

Pregunte cómo se aplica «¿Qué es la latencia?» a su organización. Basta un ejemplo concreto de su sistema actual para iniciar la conversación.

Explore nuestros servicios

welcome@engineer.company +45 50 10 61 84 Copenhague, Dinamarca

Llévese esta página

Tarjeta de contacto

Creado con esmero y cariño por D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Derechos de autor © 2025 – actualidad · Engineer ApS · Todos los derechos reservados.

Impreso desde https://engineer.company/es/glossary/latency/