Engineer

caché

un almacén de copias para responder a lecturas repetidas

Qué es

Una caché es un almacén que mantiene copias de datos cerca de quien los lee, de modo que una lectura repetida se responde desde la copia en lugar de acudir al lugar más lento donde residen realmente los datos. El plural es cachés; cada elemento almacenado es una entrada de caché, y los datos que contiene son una copia en caché.

Una caché nunca es la fuente de verdad. El archivo, la fila o la respuesta del servidor que copia conserva la autoridad, y cualquier entrada se puede descartar y volver a obtener. Eso distingue una caché de una réplica, que la replicación mantiene deliberadamente completa y sincronizada, y de un búfer, que contiene datos en tránsito que quizá todavía no existan en ningún otro lugar. La caché de páginas de Linux cumple ambas funciones: conserva los datos de archivos que se han leído y recibe primero las escrituras, marcando esas páginas como modificadas hasta que llegan al disco.

Las cachés se superponen entre los usuarios y los datos. La caché de páginas del núcleo está por debajo de cada archivo que lee un programa. PostgreSQL mantiene encima su propio conjunto de páginas, `shared_buffers`. Un navegador conserva respuestas HTTP en una caché privada para una sola persona; una CDN o un proxy inverso mantiene una caché compartida para muchas. El script de una página puede conservar respuestas en Cache Storage mediante un service worker, al margen de la caché HTTP del navegador. Y una aplicación puede guardar valores que ha calculado en un almacén de clave-valor como Redis.

El vocabulario es el mismo en todas las capas. Una lectura que encuentra su entrada es un acierto; una que no la encuentra es un fallo y continúa hasta el origen. La proporción de lecturas que aciertan es la tasa de aciertos. Cada entrada se localiza mediante su clave, que en una caché HTTP incluye como

mínimo el método de la solicitud y la URL. Cuando el almacén está lleno, descarta entradas para liberar espacio, lo que se denomina expulsión, y su política de expulsión decide cuáles elimina; la habitual es LRU, que retira la entrada que lleva más tiempo sin utilizarse.

Utilice el concepto en una decisión

Aplique «¿Qué es una caché?» a una decisión real. Podemos ayudarle a relacionar el concepto con el sistema que está construyendo o comprando. Detectar las consultas y los índices que faltan en un sistema cada año más pesado y corregir lo que realmente importa.

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. 30 días de datos eléctricos en 180 ms — 180 ms para agregar un mes de lecturas de contadores

30 días de datos eléctricos en 180 ms Póngase en contacto →

Por qué importa

Una caché suele ser la mejora de velocidad más barata que puede adquirir un sistema, más barata que hardware nuevo o un esquema nuevo. El coste llega en forma de memoria, de copias en lugares difíciles de alcanzar y de copias que dejan de reflejar la realidad. Tres decisiones sobre el almacén resuelven la mayor parte de la cuestión.

La primera es dónde se sitúa. Cuanto más cerca esté la copia de quien la lee, más ahorra: una copia en el navegador evita tanto la red como el servidor. También queda más lejos de quien la controla. Una instancia de Redis se puede vaciar con un comando; una copia en el teléfono de un visitante permanece allí bajo las reglas con las que se envió.

La segunda es el tamaño, porque la memoria que se asigna a una caché se le quita a otra cosa. La documentación de PostgreSQL sitúa el valor predeterminado habitual de `shared_buffers` en 128 MB. En un servidor dedicado con 1 GB de memoria o más, propone empezar por el 25 % y, como PostgreSQL también depende de la caché del sistema operativo, indica que es poco probable que asignar más del 40 % funcione mejor que una cantidad menor. Dos cachés comparten una máquina.

La tercera es quién comparte el almacén. RFC 9111 distingue una caché privada, para un usuario, de una compartida, que reutiliza las respuestas que guarda para más de un usuario. Una respuesta personal en un almacén compartido es la

forma en que la página de una persona llega a otra, por lo que una caché compartida no debe reutilizar una respuesta a una solicitud que incluía una cabecera `Authorization`, salvo que la respuesta lo permita expresamente.

De las tres se desprende una regla: debe ser seguro vaciar una caché. Si al vaciarla se pierden datos, se ha convertido en una base de datos sin las garantías de una base de datos. Cuánto tiempo sigue siendo una entrada lo bastante válida para servirla y qué la retira antes de tiempo son cuestiones del almacenamiento en caché, la práctica; el cometido del almacén es guardar copias, encontrarlas por su clave y ceder espacio cuando escasea.

Cómo se aplica en la práctica

En un servidor con 464 MB que ejecuta su forja y su servidor web, la caché de páginas del núcleo determina qué debe medir una alarma. Un interruptor de hombre muerto que da la alarma al dejar de emitir señales se ejecuta allí cada quince minutos, con una variación aleatoria del intervalo: un script mide la memoria disponible y el espacio libre en disco y envía un ping a un servicio externo solo mientras ambos superan sus umbrales. La memoria disponible es la medida correcta, porque en un equipo Linux sano la memoria libre ronda cero: el núcleo utiliza la memoria ociosa como caché. El script leía la columna correcta, pero seis nombres de variables a su alrededor decían «free» («libre»). Quien confiara en esos nombres ante un umbral del diez por ciento vería doce megabytes libres en 464 MB, daría el control por defectuoso y cambiaría un solo carácter, dejando una alarma que permanece activada y se silencia en el plazo de una semana. Desde entonces se han corregido los nombres.

Una capa por encima, una caché de aplicación se sitúa delante de una base de datos. La nota sobre optimizar una base de datos antes de fragmentarla coloca una delante de una situación de existencias agotadas. Cuando todas las unidades de un artículo están reservadas, la base de datos puede indicar cuándo caduca la primera reserva, diez minutos después de haberse creado, y la nota guarda «totalmente reservado» en caché con exactamente esa caducidad. Hasta entonces, es la caché la que rechaza a los visitantes, no la base de datos. Una reserva liberada antes de tiempo, por una tarjeta rechazada o un carrito abandonado, elimina la entrada, de modo que la unidad vuelve a estar a la venta inmediatamente. La caché puede rechazar; solo la base de datos vende, mediante un `UPDATE` condicional. Una entrada desactualizada puede impedir una venta durante unos segundos y nunca puede vender dos veces una misma unidad.

La capa más externa es el propio dispositivo del visitante, donde este sitio no crea ningún almacén gestionado mediante scripts. Su página sobre cookies indica que no utiliza IndexedDB ni Cache Storage ni instala ningún service worker. El navegador puede seguir conservando respuestas en su caché HTTP habitual, separada de una copia sin conexión gestionada por el sitio.

Dónde surgen los problemas

Tras un reinicio, el origen recibe todas las lecturas a la vez. Una caché fría falla en todas las búsquedas, por lo que un despliegue, un reinicio de la caché o una conmutación por error envía toda la carga de lectura a la base de datos. Un sistema que solo aguantaba porque la mayoría de las lecturas eran aciertos puede caer antes de que la caché se vuelva a llenar. Dimensione el origen para la ruta de los fallos de caché, precargue la caché antes de que reciba tráfico y devuelva el tráfico por etapas. `pg_prewarm` de PostgreSQL lo hace con la caché de búferes y puede registrar qué contenían los búferes compartidos y volver a leerlo tras un reinicio.

La caché crece hasta agotar la memoria de la máquina. En sistemas de 64 bits, Redis no establece ningún límite de memoria de forma predeterminada, de modo que una instancia utilizada como caché crece con su conjunto de claves hasta que interviene el sistema operativo. Establezca a la vez un límite y una política. Con `noeviction`, una instancia llena responde con un error a los comandos que almacenarían datos nuevos en caché: encaja con un almacén que debe conservarlo todo y sorprende en uno pensado para olvidar. La documentación de Redis considera `allkeys-lru` un buen valor predeterminado si no hay motivos para preferir otro:

```
maxmemory 256mb
maxmemory-policy allkeys-lru
```

Un visitante recibe una página en el idioma equivocado. La clave omite una entrada de la que depende el valor. Una página generada en danés se almacena solo bajo su URL, y el siguiente lector recibe danés, sea cual sea el idioma que haya solicitado. Construya la clave con todas las entradas que cambian la salida, como el idioma, la moneda o el rol. Para las cachés HTTP, la cabecera de respuesta `Vary` indica las cabeceras de solicitud que forman parte de la clave, y RFC 9111 solo permite reutilizar una respuesta almacenada cuando esas cabeceras coinciden o después de revalidarla.

La aplicación instalada sigue ejecutando código antiguo. Las entradas de Cache Storage no tienen una caducidad propia y sobreviven a una actualización del script del service worker. Los archivos modificados bajo el mismo nombre de caché se siguen sirviendo desde las copias antiguas, y cambiar el nombre de la caché en cada publicación sin eliminar las anteriores deja una más en el dispositivo cada vez. Dé a cada caché un nombre con versión, como recomienda la especificación de Service Workers, y elimine todas las anteriores cuando el nuevo worker tome el control.

Lecturas adicionales

- RFC 9111: almacenamiento en caché HTTP — el estándar del IETF para cachés HTTP privadas y compartidas, sus claves y aquello que una caché compartida debe dejar intacto.
- Núcleo Linux: conceptos de gestión de memoria — cómo la caché de páginas conserva datos de archivos en memoria y por qué sus páginas se consideran recuperables.
- PostgreSQL: consumo de recursos — `shared_buffers`, su valor predeterminado habitual y las recomendaciones de dimensionamiento que tienen en cuenta la caché del sistema operativo.
- Service Workers — la especificación del W3C sobre Cache Storage y en qué se diferencia de la caché HTTP del navegador.
- Redis: expulsión de claves — `maxmemory` y lo que hace una caché llena con cada política de expulsión.

Glosario

Elija según sus restricciones reales

Traiga un conjunto de datos representativo, las preguntas que debe responder y el cuello de botella actual. Podemos acordar cómo medir la calidad, la capacidad de procesamiento y la recuperación.

Vea cuánto cuesta Póngase en contacto →

Haga práctica la terminología

Pregunte cómo se aplica «¿Qué es una caché?» 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/cache/