Introducción

Memcached es un sistema distribuido de propósito general para caché basado en memoria, diseñado por Danga Interactive y que es muy usado en la actualidad por múltiples sitios web.

Memcached es empleado para el almacenamiento en caché de datos u objetos en la memoria RAM, reduciendo así las necesidades de acceso a un origen de datos externo (como una base de datos o una API). Memcached tiene versiones para Linux, Windows y MacOS y se distribuye bajo licencia de software libre permisiva.

Su funcionamiento se basa en una tabla hash distribuida a lo largo de varios equipos. Conforme ésta se va llenando, los datos que más tiempo llevan sin ser utilizados se borran para dar espacio a los nuevos. Normalmente, las aplicaciones comprueban primero si pueden acceder a los datos a través de Memcached antes de recurrir a un almacén de datos más lento, como puede ser una base de datos.

Este sistema es empleado por varios de los sitios más activos y visitados de la red, como YouTube, Reddit,​ Playdom,​ Zynga,​ Facebook y Twitter. Heroku ofrece un servicio de Memcached gestionado con NorthScale​ como parte de su PaaS. Google App Engine ofrece también un servicio de memcached​ a través de un API.

Memcached es una herramienta de desarrollo, no un "acelerador de código", ni tampoco es un middleware de base de datos. Si está intentando configurar una aplicación que ha descargado o comprado para usar memcached, lea la documentación de su aplicación.

Funcionalidad

memcached le permite tomar memoria de partes de su sistema donde tiene más de lo que necesita y hacerla accesible a áreas donde tiene menos de lo que necesita.

memcached también le permite hacer un mejor uso de su memoria. Si considera el diagrama a la derecha, puede ver dos escenarios de implementación:

1. Cada nodo es completamente independiente (arriba).
2. Cada nodo puede hacer uso de la memoria de otros nodos (parte inferior).

El primer escenario ilustra la estrategia de implementación clásica, sin embargo, encontrará que es un desperdicio en el sentido de que el tamaño total del caché es una fracción de la capacidad real de su granja web, pero también en la cantidad de esfuerzo requerido para mantener el caché. consistente en todos esos nodos.

Con memcached, puede ver que todos los servidores están buscando en el mismo grupo virtual de memoria. Esto significa que un elemento determinado siempre se almacena y siempre se recupera desde la misma ubicación en todo su grupo web.

Además, a medida que la demanda de su aplicación crece hasta el punto en que necesita tener más servidores, generalmente también crece en términos de los datos a los que se debe acceder regularmente. Una estrategia de implementación donde estos dos aspectos de su sistema se escalan juntos simplemente tiene sentido.

La ilustración de la derecha solo muestra dos servidores web por simplicidad, pero la propiedad sigue siendo la misma a medida que aumenta el número. Si tuviera cincuenta servidores web, aún tendría un tamaño de caché utilizable de 64 MB en el primer ejemplo, pero en el segundo, tendría 3,2 GB de caché utilizable.

Por supuesto, no es necesario que use la memoria de su servidor web para el caché. Muchos usuarios de memcached tienen máquinas dedicadas que están diseñadas para ser solo servidores de memcached.

Casos de uso de Memcached destacados

**Almacenamiento en caché**

Memcached es una excelente opción para implementar un alto rendimiento del almacenamiento en caché en la memoria para disminuir la latencia de acceso a los datos, aumentar la capacidad y aligerar la carga de sus sistemas de backend. Memcached puede servir elementos en caché en menos de un milisegundo y le permite realizar escalaciones fácilmente y de forma rentable en cargas mayores. Memcached está pensado para el almacenamiento en caché de resultados de consultas de bases de datos, de sesiones, de API y de elementos como imágenes, archivos y metadatos.

**Almacén de sesiones**

Memcached es una opción para el almacenamiento de datos en la memoria muy utilizada entre los desarrolladores de aplicaciones y que sirve para almacenar y gestionar datos de sesiones para aplicaciones de Internet en los casos en los que la persistencia no es algo fundamental. Memcached está diseñado para proporcionar escalas y latencias inferiores a un milisegundo requeridas para la gestión de datos de sesión, como perfiles de usuario, credenciales y estados de sesión.

Arquitectura

El sistema usa una arquitectura cliente-servidor. Los servidores mantienen un array asociativo clave-valor; los clientes añaden datos al array y acceden a él. Las claves pueden tener una longitud de hasta 250 bytes y los datos pueden tener un tamaño de hasta 1 megabyte.

Los clientes usan librerías cliente para acceder a los servidores que, por defecto, utilizan el puerto 11211 Cada cliente mantiene una lista de todos los servidores; los servidores no se comunican entre ellos. Si un cliente desea establecer o leer el valor correspondiente a cierta clave, la librería cliente primero hace un cálculo mediante un algoritmo hash para determinar el servidor que va a utilizar. Entonces se pone en contacto con el servidor y éste usará otro hash para determinar dónde almacenar o leer el valor correspondiente.

El servidor mantiene los valores en RAM. Si un servidor agota su memoria, descarta los valores más antiguos. Por tanto, los clientes deben de tratar Memcached como una caché transitoria; no pueden asumir que los datos almacenados en Memcached estarán ahí cuando los necesiten. Un producto compatible a nivel de protocolo con Memcached llamado MemcacheDB proporciona almacenamiento permanente. Hay también una solución llamada Membase de Northscale que proporciona persistencia, replicación y clustering.

Para que un cliente pueda leer los datos almacenados por otro cliente, deberían ambos usar el mismo algoritmo hash para localizar los servidores.

Un despliegue típico tendría varios servidores y muchos clientes. Sin embargo, es posible usar Memcached en un único ordenador, actuando simultáneamente como cliente y servidor.

Seguridad

La mayor parte de los despliegues de Memcached se hacen en redes corporativas "de confianza", donde los clientes se pueden conectar libremente a cualquier servidor. Hay casos, sin embargo, en los que Memcached es desplegado en redes inseguras o en las que los administradores querrían ejercer un control en la forma que se conectan los cliente. Para este propósito Memcached puede ser compilado con el soporte opcional de autentificación SASL. El soporte SASL requiere el uso del protocolo binario.

Este sistema está pensado para ser usado en sistemas que no están conectados a Internet, pero por desgracia muchos de ellos sí lo están. Por ello han rcibido ataques.

Pruebas

Memcached funciona en la mayoría de los sistemas Linux y BSD. No hay soporte oficial para las compilaciones de windows.