arquitectura de php

Erick Julian Latorre

Contenido

[1 cómo convertir el código php en bytecode? 1](#_Toc180097309)

[2 citar ejemplos de php a bytecode. 1](#_Toc180097310)

2.1[Ejemplo 1](#_Toc180097311)

[3 cómo funciona el compilador JIT? 1](#_Toc180097312)

3.1[(Just-In-Time): 1](#_Toc180097313)

[4.Que es un servidor web. Arquitectura del servidor Web? 2](#_Toc180097314)

4.1[Cliente (navegador web): 2](#_Toc180097315)

4.2[Servidor web: 2](#_Toc180097316)

4.3[Aplicación web (opcional): 2](#_Toc180097317)

4.4[Base de datos (opcional): 2](#_Toc180097318)

[5 Investigar y alistar las diferentes extensiones de PHP con su descripción 2](#_Toc180097319)

## 1 cómo convertir el código php en bytecode?

## Para convertir código PHP en bytecode, primero asegúrate de tener habilitado Opcache en tu archivo php.ini, lo que optimiza el rendimiento almacenando el bytecode. Para visualizar el bytecode, puedes usar la extensión vld, que se instala con PECL y se agrega a tu configuración de PHP. Al ejecutar tu script, vld mostrará la representación del bytecode. Alternativamente, puedes usar HHVM, que compila el código PHP a bytecode al ejecutarlo. Ambas opciones te permiten analizar el bytecode generado, aunque de formas diferentes.

## 2 citar ejemplos de php a bytecode.

### Ejemplo

Asi se muestra el código en el compilador HHVM

hhvm -v Eval.Jit=true script.php

compila script.php a bytecode y lo ejecuta.

HHVM actúa como un compilador Just-In-Time (JIT) que traduce PHP a un bytecode intermedio que puede ejecutarse más rápido mejora el rendimiento del código PHP al reducir el tiempo de ejecución en futuras solicitudes.

## 3 cómo funciona el compilador JIT?

### (Just-In-Time):

El **JIT** es una técnica utilizada por PHP a partir de la versión 8. Lo que hace es compilar ciertas partes del código PHP en tiempo de ejecución directamente en código de máquina. En lugar de interpretar el código línea por línea, el JIT convierte partes del código a instrucciones más eficientes (nativas de la CPU) y las ejecuta más rápido. Esto mejora el rendimiento en aplicaciones muy intensivas en cálculo, como juegos o procesamiento de datos

## 4.Que es un servidor web. Arquitectura del servidor Web?

Un **servidor web** es un software que maneja solicitudes HTTP/HTTPS y entrega contenido como páginas HTML, imágenes, y otros archivos a los navegadores que las solicitan. Ejemplos comunes son Apache, Nginx y IIS. La **arquitectura del servidor web** incluye los siguientes componentes:

### Cliente (navegador web):

Envía una solicitud HTTP al servidor web.

### Servidor web:

Recibe la solicitud, la procesa y devuelve el recurso solicitado (una página web, un archivo, etc.).

### Aplicación web (opcional):

Si la solicitud requiere procesamiento dinámico (por ejemplo, PHP o Python), el servidor web envía la solicitud a la aplicación para que la procese.

### Base de datos (opcional):

La aplicación puede conectarse a una base de datos para recuperar o almacenar datos según sea necesario.

## 5 Investigar y alistar las diferentes extensiones de PHP con su descripción

* **SOAP**: Permite trabajar con servicios web basados en el protocolo SOAP (Simple Object Access Protocol). Ideal para integraciones con sistemas que utilizan SOAP.
* **Imagick:** Una interfaz para la biblioteca ImageMagick, que permite crear, editar y convertir imágenes en diversos formatos. Es muy poderosa para manipulación avanzada de imágenes.
* **YAML**: Permite leer y escribir archivos YAML (YAML Ain't Markup Language), un formato de serialización de datos muy utilizado en configuraciones.
* **Xdebug:** Herramienta de depuración y análisis de rendimiento para PHP. Es esencial para los desarrolladores que necesitan rastrear y solucionar problemas en su código.
* **New Relic**: Ofrece monitoreo y análisis de rendimiento para aplicaciones PHP, ayudando a identificar cuellos de botella y optimizar el rendimiento.
* **Phalcon:** Un framework PHP que se implementa como una extensión en C, ofreciendo un rendimiento muy alto y menor uso de recursos.
* **PDO (PHP Data Objects**): Proporciona una interfaz consistente para acceder a múltiples bases de datos. Permite preparar y ejecutar sentencias SQL con facilidad y seguridad.
* **GD**: Una librería para crear y manipular imágenes en varios formatos, como JPEG, PNG y GIF. Ideal para generar gráficos o procesar imágenes.
* **cURL**: Permite realizar solicitudes HTTP desde tu código PHP. Es útil para interactuar con APIs y otros servicios web.
* **mbstring**: Proporciona funciones para manejar cadenas con codificación multibyte, lo cual es esencial para trabajar con textos en varios idiomas.
* **OpenSSL**: Ofrece funciones para trabajar con SSL y TLS, incluyendo generación y verificación de firmas digitales y cifrado de datos.
* **XML**: Incluye varias herramientas para manejar y manipular XML. Muy útil para trabajar con datos estructurados en este formato.
* **Memcached**: Una extensión que proporciona una interfaz a la herramienta Memcached, útil para caching en aplicaciones web, mejorando el rendimiento mediante el almacenamiento en memoria de datos frecuentemente accedidos.
* **APCu** (Alternative PHP Cache): Una solución de cacheo para PHP que almacena datos en la memoria compartida, mejorando la velocidad de la aplicación mediante el cacheo de datos y código.