Decisiones de diseño:

Latencia: Tiempo que demora la información en viajar de un punto de la red a otro [1]. La latencia se ve afectada por la posición geográfica y la calidad de las redes entre los servidores y los usuarios. También se puede reducir la latencia configurando los servidores o también mejorando la infraestructura de alojamiento de estos.

Ancho de banda: Es una medida del rendimiento [1].

Rendimiento: Eventos por unidad de tiempo. Ejemplos de eventos pueden ser:

Número de trabajos completados, cantidad de invocaciones remotas, cantidad de datos transmitidos.

Apache Hadoop es un framework de software que soporta aplicaciones distribuidas bajo una licencia libre, permite que las aplicaciones trabajen con miles de nodos y petabytes (1015 bytes). [2]

Que optimizar en un sistema distribuido:

Se optimiza la latencia por defecto.

El sistema de archivos distribuidos de hadoop es un sistema de archivos distribuido, escalable y portátil escrito en java para el framework de hadoop. [2, 3]

Procesamiento por lotes: Es la ejecución de un programa sin el control o supervisión directa de un usuario, generalmente usado para tareas repetitivas sobre grandes conjuntos de información. [6]

Red Ad-hoc inalámbrica: Es un tipo de red inalámbrica, descentralizada, donde cada nodo participa en el encaminamiento mediante el reenvío de datos hacia otros nodos. Pueden usar la inundación (flooding) de la red para el reenvío de datos [7].

AWS snowball: Es una solución de transporte de datos que acelera la transferencia desde terabytes a petabytes de datos desde Amazon Web Services (AWS). [8]

Consistencia requiere de consenso y acuerdo.

No siempre es necesario un 100% de consistencia, se puede mejorar la latencia al reducir la consistencia.

Modelos de consistencia:

Consistencia casual: Escrituras que están potencialmente relacionadas de forma casual deben ser vistas por todos los procesos en el mismo orden. [9]

Consistencia secuencial: Forma ligeramente más débil de la consistencia estricta. El resultado de la ejecución es el mismo si las operaciones de todos los procesos sobre el dato fueron ejecutadas en algún orden secuencial. [9]

Consistencia débil: Los accesos a variables de sincronización asociadas con los datos almacenados son secuencialmente consistentes. [9]

Teorema CAP: Consistencia, disponibilidad y tolerancia particiones, Consistencia se refiere a que todos los nodos ven la misma data al mismo tiempo, Disponibilidad se refiere a una garantía de que todos los requerimientos reciban una respuesta de que el requerimiento fue exitoso o fallido, y tolerancia a partición se refiere a que el sistema sigue funcionando a pesar de la pérdida arbitraria de mensajes, o la falla de parte del sistema. El teorema dice que podemos asegurar 2 de 3 de estas características en un sistema distribuido, pero nunca las 3.

Apache storm: Procesa datos en tiempo real y la entrada normalmente proviene de un sistema de colas de mensajes. [10]

Cluster: Conjuntos o conglomerados de ordenadores unidos entre si por una red de alta velocidad y se comportan como si fueran una sola computadora.

Arquitectura de amazon DynamoDB:

Es una arquitectura verdaderamente distribuida.

Los datos son difundidos a través de servidores llamados nodos de almacenamiento.

La aplicación del cliente puede conectarse usando uno de los 2 enfoques descritos a continuación:

Enrutamiento usando balanceo e carga.

Librería del cliente que refleja el esquema de particiones de dynamo y puede determinar el host de almacenamiento a conectar.

Ventajas de balanceo de carga: No se requiere de un código específico de dynamo en el lado del cliente.

Ventaja de librería del cliente: Ahorramos 1 salto de red para balanceo de carga.

La replicación sincrónica no se puede alcanzar en requerimientos de alta velocidad y escalabilidad en amazon.

DynamoDB está diseñado para ser una solución de almacenamiento siempre grabable.

Permite múltiples versiones de datos en múltiples nodos de almacenamiento.

Los conflictos de resolución ocurren durante la lectura y no durante la escritura, pueden darse:

Resolución de conflicto sintáctico.

Resolución de conflicto semántico.

Dynamo es una base de datos NoSQL de amazon, es de código cerrado. [11]

Dynamo presenta disponibilidad y escalabilidad.

El mecanismo de búsqueda y una gran distribución de los datos entre los nodos juega un papel importante en el tiempo de respuesta, que no debe ser mayor a 300 ms

Saltos de red: Un salto es una parte del camino entre la fuente y su destino. Cada vez que los paquetes se pasan al siguiente dispositivo de red se produce un salto.

Protocolo sin estado: Es un protocolo de comunicaciones que trata cada petición como una transacción independiente que no tiene relación con cualquier otra solicitud anterior, no se necesita mantener la información del estado o de la sesión. El estado puede ser cualquier información con respecto a las interacciones en curso. Sus ventajas son: Simplifica el diseño del servidor porque no hay necesidad de asignar dinámicamente almacenamiento para tratar las conversaciones en curso. Una ventaja es que si un cliente desaparece en medio de la transmisión ninguna parte del sistema tiene que ser responsable de limpiar el estado actual del servidor. Una desventaja es que puede ser necesario incluir información adicional en cada petición que necesitará ser interpretada por el servidor. HTTP es un protocolo sin estado. [13]

Protocolo con estado: Se refiere a registrar, utilizando al servidor, los resultados de las operaciones entre un cliente y un servidor (si la transacción fue una petición única seguida de una respuesta o si se realizó una serie de intercambio de datos), se realiza un seguimiento de la información que llega de una transmisión entre cliente y servidor, en esta se trata a cada paquete como parte de una serie. Un ejemplo es FTP. [14]

Las ventajas de protocolos con estados son desventajas en protocolos sin estado y viceversa.

Estado blando (Soft state): Es un estado útil para la eficiencia, pero no es esencial, ya que puede ser regenerado o remplazado si es necesario.

Aspectos de diseño para protocolos con estado:

Timeouts (tiempo de espera).

Pedidos y respuestas duplicados.

Idempotencia: Operaciones que producen los mismos resultados si se ejecuta varias veces. [15]

Crash: Condición en la que una aplicación deja de funcionar de la forma esperada y dejan de responder a otras partes del sistema.

Los problemas relativos a las regiones críticas, exclusión mutua y la sincronización generalmente se resuelven con semáforos y monitores, pero estos no aplican a sistemas distribuidos.

La sincronización de relojes no tiene que ser absoluta en SO, para lograrse se emplean algoritmos.

Los relojes pueden ser lógicos o físicos.

Relojes lógicos: Circuito en las computadoras para el registro del tiempo, este se llama dispositivo de reloj.

Relojes físicos: Se usa el algoritmo de Lamport para obtener un orden de eventos sin ambigüedades, los valores de tiempo asignados no tienen porque ser cercanos a los tiempos reales que ocurren.

En la sincronización de relojes, si una máquina tiene un receptor UTC, todas las máquinas deben sincronizarse con ella, en caso de que ninguna máquina tenga dicho receptor, todas deben llevar el registro de su propio tiempo y todas tener un tiempo aproximado muy cercando, lo mayor que sea posible.

La sincronización garantiza que los procesos se ejecutarán en forma cronológica además de respetarse el orden de los eventos dentro del sistema. [17]

Algoritmo para la sincronización de relojes: Un algoritmo para llevar a cabo esta tarea se basa en el uso del tiempo coordenado universal (UTC) que es recibido por un equipo en el sistema, el que lo recibe se conoce como receptor de UTC, también recibe solicitudes a las cuales responde en el menor tiempo posible informando el tiempo UTC solicitado, con lo cual todas las máquinas del sistema pueden actualizar su hora y se mantenga sincronizado todo el sistema. Un gran problema es que el tiempo no puede correr hacia atrás, pero el tiempo del receptor no puede ser menor que el tiempo de la máquina que le solicitó el tiempo, se deben procesar las solicitudes involucrando también el tiempo de las interrupciones (en caso de haberlas), el intervalo del tiempo de transmisión de la solicitud y su respuesta debe ser tomando en cuenta en el proceso de sincronización. [17]

Algoritmo de Cristian: El sistema maneja su propia hora, realiza un muestreo periódico de todas las máquinas del sistema y establece un tiempo promedio para realizar la sincronización. [17]

Algoritmo con promedio: Cada máquina del sistema informa su hora actual con cada mensaje que requiera enviar a otras máquinas del sistema e inicializa localmente un cronómetro, cada máquina promedia su hora local con el uso de las horas que informan el resto de las máquinas. [17]

Manifiesto reactivo:

Debido a que en la actualidad se requieren aplicaciones que soporten más carga de información sin afectar el tiempo de respuesta de los usuarios. El sistema debe tener una arquitectura fundamentalmente asincrónica y orientada a mensajes. [18]

Responsivo (Responsive): Se debe detectar rápidamente los problemas y tratarlos efectivamente para mejorar la usabilidad por parte del usuario. Estos sistemas se enfocan en proveer tiempos de respuesta rápidos y consistentes para mejorar la calidad del servicio y así se simplifica el tratamiento de errores, mejora la confianza con el usuario final. El sistema debe responder aun cuando ocurran errores. [18]

Resilientes (Resilient): Un sistema debe mantenerse responsivo cuando encuentre fallas, para hacerlo, se emplean estrategias de replicación, contención, aislamiento y delegación. Si falla un componente del sistema debe ser aislado del resto para asegurar que se podrá recuperar sin comprometer el resto del sistema. Un componente externo se encarga de la recuperación del componente aislado, se asegura alta disponibilidad aplicando replicación donde se crea necesario. El cliente no es obligado a manejar las fallas que cada componente genere. [18]

Elásticos (Elastic): Un sistema se mantiene responsivo bajo diferentes cargas de trabajo o cambios en la frecuencia de peticiones incrementando o reduciendo los recursos asignados para servir dichas peticiones. Debemos diseñar sistemas sin cuellos de botella, también diseñar puntos de contención que repliquen componentes y balancear carga entre ellos. Deben soportar algoritmos predictivos y de escalabilidad para ser escalables de forma rentable. [18]

Orientados a mensajes (Message driven): Se envían mensajes asincrónicos entre componentes, asegurando bajo acoplamiento, aislamiento, transparencia e independencia de su ubicación, esto asegura la administración de la carga y elasticidad a través del monitoreo de colas en el sistema. [18]

Algoritmo no determinista: Algoritmo que, con la misma entrada, ofrece muchos resultados, su solución no es única. [19]

Tipos de bloqueos en SO:

Bloqueo de comunicación: Cuando un proceso intenta enviar un mensaje a varios procesos.

Bloqueo de recursos: Cuando procesos pelean por el acceso a dispositivos de E/S.

Algoritmos para manejo de interbloqueos: [20]

Algoritmo del avestruz: Ignorar el problema.

Algoritmo de detección: Permitir que ocurran interbloqueos, detectarlos y recuperarse de ellos.

Algoritmo de prevención: Lograr que los interbloqueos no puedan ocurrir desde el punto de vista estructural.

Algoritmo de evasión: Lograr que los interbloqueos no puedan ocurrir mediante la asignación de recursos.

Se recomienda:

Evitar estados compartidos en implementaciones multihilos.

No bloquearse.

Particionar y replicar.

Replicación de datos: [21]

Varias copias de la información para mejorar rendimiento, disponibilidad y tolerancia a fallos.

Particionar datos: [22]

Es necesario en ciertas ocasiones particionar una solución en varias más pequeñas, solo con un subconjunto de las aplicaciones en forma flexible de organizar los proyectos para su desarrollo.

Shardding (trozear, fragmentar): Es dividir una base de datos en otras más pequeñas, que son más manejables.

Single shard: Cuando tengo un solo fragmento con los datos. Cada fragmento actua como fuente única para este subconjunto de datos.

Multiple shards: Los datos están fragmentados en distintas máquinas.

Arquitectura share nothing (SN): Arquitectura distribuida en la que cada nodo es independiente y autosuficiente, no tiene un único punto de contención en todo el sistema.

Acoplamiento (bajo acoplamiento): Dependencia entre los diferentes elementos de la arquitectura SOA o las dependencias con los consumidores.

Comunicación asincrónica: Comunicación que se establece entre personas de manera diferida en el tiempo, es decir, cuando no hay coincidencia temporal.

Comunicación sincrónica: Intercambio de información por internet en tiempo real. Es un concepto que se enmascara dentro de la CMC (computer mediated comunication), que es aquel tipo de comunicación que se da entre personas y que está mediatizada por ordenadores.

Transparencia de localización: Es el uso de nombres para identificar los recursos de la red, en lugar de su ubicación real.

Política LRU: Se basa en el principio de localidad temporal, si un bloque ha sido utilizado recientemente es probable que sea utilizado en un futuro cercano, el mejor candidato a remplazar es el bloque que no ha sido utilizado por más tiempo.

Política LIFO: El bloque que entra en el último puesto de la pila es el principal candidato para ser remplazado, tiene mal rendimiento porque se opone al principio de localidad temporal.

IPC (): Es un formato de mensaje ligero utilizado para comunicarse entre cachés web, se usa para intercambiar pistas acerca de la existencia de URL en caches vecinos.

Red de entrega de contenidos (CDN): Red superpuesta de computadoras que contienen copias de datos, colocados en varios puntos de la red para maximizar el ancho de banda, con el fin de evitar embudos en el servidor principal.

Redes definidas por software: Son un conjunto de técnicas relacionadas con el área de redes computacionales, cuyo objetivo es facilitar la implementación e implantación de servicios de red de una manera determinística, dinámica y escalable.

[1] <https://www.xataka.com/basics/que-son-el-ping-y-la-latencia-y-por-que-no-solo-importa-la-velocidad-en-tu-conexion>

[2] <https://es.wikipedia.org/wiki/Hadoop#Arquitectura>

[3]<https://translate.google.com.ec/translate?hl=es&sl=en&u=https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html&prev=search>

[4]<http://blog.iweb.com/es/2014/02/entender-analizar-reducir-latencia/2463.html>

[5] <http://www.ttandem.com/blog/la-importancia-de-mejorar-la-velocidad-de-tu-sitio-web/>

[6] <https://es.wikipedia.org/wiki/Procesamiento_por_lotes>

[7] <https://es.wikipedia.org/wiki/Red_ad_hoc_inal%C3%A1mbrica>

[8] <https://aws.amazon.com/es/snowball/faqs/>

[9] <http://sistemasoperativosdistribuidoss.blogspot.com/2012/05/modelos-de-consistencia.html>

[10]<http://www.w3ii.com/es/apache_storm/apache_storm_distributed_messaging_system.html>

[11] <http://www.nosql.es/blog/nosql/amazon-dynamo.html>

[12]<https://translate.google.com.ec/translate?hl=es&sl=en&u=https://en.wikipedia.org/wiki/Hop_(networking)&prev=search>

[13] <https://es.wikipedia.org/wiki/Protocolo_sin_estado>

[14] <http://www.ordenador.online/Redes/Otros-Inform%C3%A1tica-Redes/%C2%BFQu%C3%A9-es-el-Protocolo-Stateful-.html>

[15] <https://es.wikipedia.org/wiki/Idempotencia_(inform%C3%A1tica)>

[16] <https://es.wikipedia.org/wiki/Crash_(inform%C3%A1tica)>

[17] <http://www.monografias.com/trabajos55/sincronizacion-sistemas-distribuidos/sincronizacion-sistemas-distribuidos.shtml>

[18] <https://code2read.com/2015/10/10/sistemas-distribuidos-el-manifiesto-de-sistemas-reactivo/>

[19] <https://es.wikipedia.org/wiki/Algoritmo_no_determinista>

[20] <https://es.slideshare.net/yramirez/bloqueos-en-sistemas-distribuidos-presentation>

[21] <http://web.dit.upm.es/~joaquin/so/replic/replic.pdf>

[22] <https://msdn.microsoft.com/es-es/library/ms185282(v=vs.90).aspx>

<https://en.wikipedia.org/wiki/Shard_(database_architecture)>

<http://oa.upm.es/32647/1/PFC_FERNANDO_GODINO_GONZALEZ.pdf>

<https://es.wikipedia.org/wiki/Arquitectura_shared_nothing>

<http://www.jaruiz.es/blog/tag/loose-coupling/>

<https://prezi.com/oeyplxbxuxy2/diferencias-entre-comunicacion-sincronica-y-asincronica/>

<http://eprints.ucm.es/16712/1/Memoria_Roberto_Rodriguez_Rodriguez.pdf>

<https://translate.google.com.ec/translate?hl=es&sl=en&u=http://www.networksorcery.com/enp/protocol/icp.htm&prev=search>

<https://es.wikipedia.org/wiki/Redes_definidas_por_software>