Bases de datos distribuidas

¿Qué es una base de datos distribuida?

Una base de datos distribuida es una base de datos que corre y almacena los datos en múltiples computadoras.

Estas bases de datos operan en dos o más servidores interconectados a una red.

A cada una de estas bases de datos se las conoce como instancias o nodos

Características

- **Independiente de la ubicación**.
  - Los datos físicos almacenados en múltiples sitios y manejados por DBMS independientes
- **Procesamiento de queries distribuidos**
  - Los datos se distribuyen entre las distintas PCs lo que en algunos casos puede ser bastante demandante si no se está sobre una red de alta velocidad
- **Manejo de transacciones distribuidos**
  - Consistencia a través de protocolos de commit, técnicas de control de concurrencia distribuidas y métodos de recuperación distribuidos
- Se representan como una única base de datos lógica interconectadas
- Todas las bases de datos en una colección están conectadas por una red y se comunican entre ellas

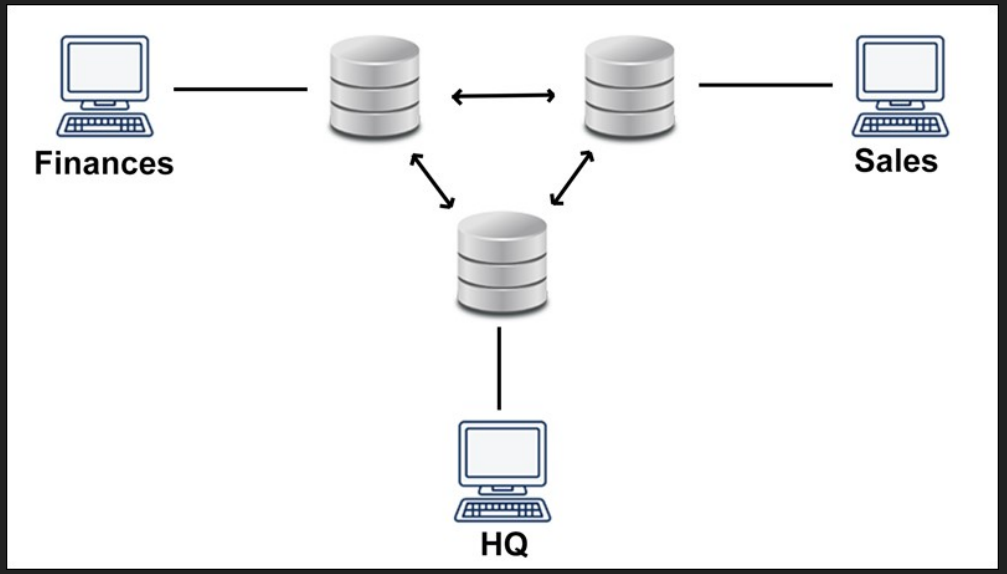
Tipo - Homogénea

Todos los sitios usan un sistema operativo y SGBD idéntico y la estructura de datos es la misma en todos los sitios.

Sus propiedades son:

- Todos los sitios usan software similar
- Todos los sitios usan un SGBD idéntico o del mismo vendedor
- Cada sitio conoce del otro y cooperan para procesar los pedidos del usuario
- Se accede a través de una interface como si fuera una sola base de datos

Tipo - Homogénea



Tipo - Homogénea

- **Autónoma**
  - Cada base de datos es autónoma que funciona por sí sola. Están integradas por una aplicación que maneja todo y utilizan mensajes para compartir los datos que son actualizados.

- **No-Autónoma**
  - Los datos son distribuidos a través de todos los nodos y un SGBD central o maestro coordina la actualización de datos a través de todos los sitios

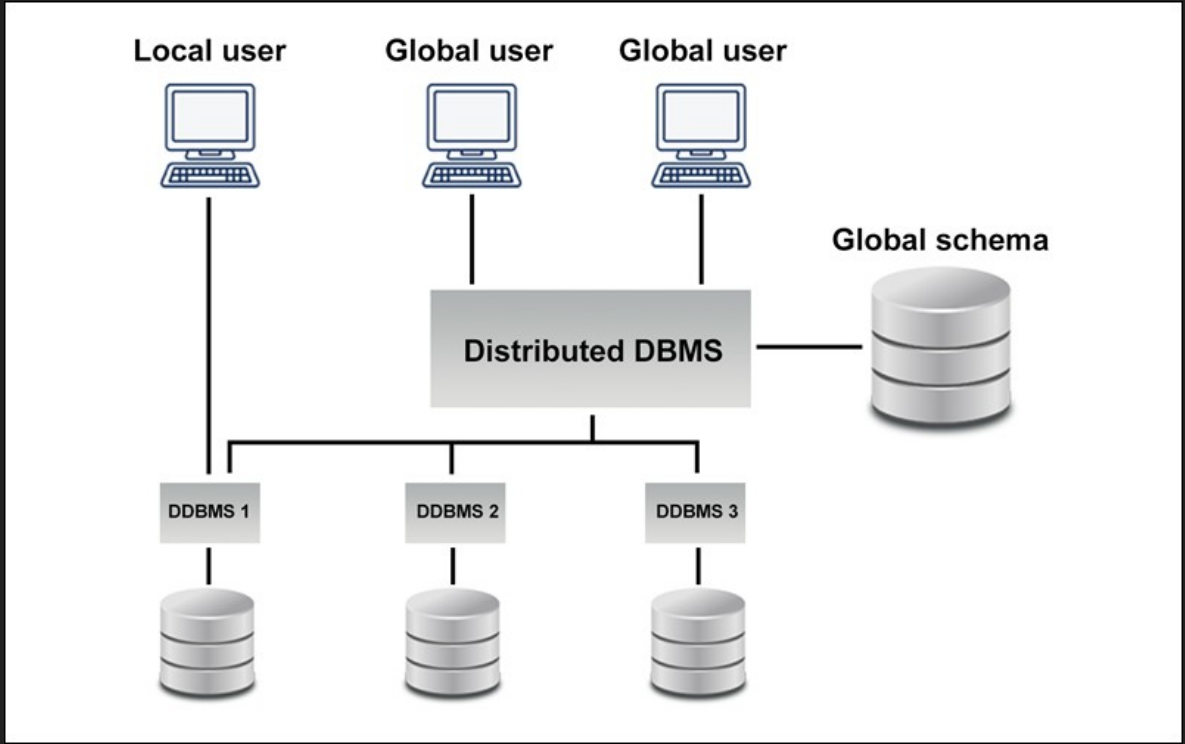
Tipo - Heterogénea

Todos los sitios tiene diferentes sistemas operativos, SGDB y modelos de datos

Sus propiedades son:

- Todos los sitios usan un software y schema diferente
- Los sitios pueden utilizar distintos SGDB como relacionales, jerárquicos o orientados a objetos
- El procesamiento de queries es complejo debido a que los schemas son distintos
- El procesamiento de transacciones puede ser complejo debido a que los softwares son distintos
- Los sitios pueden no conocerse entre ellos por lo que la cooperación es limitada

Tipo - Heterogénea



Tipo - Heterogénea

- **Federated**
  - Los sistemas son naturalmente independientes e integrados para que funcionen como una sola base de datos

- **Un-Federated**
  - Los sistemas de bases de datos utilizan un módulo de coordinación centralizado a través del cual las bases de datos son accedidas.

Replicación

La replicación me permite almacenar copias de datos en diferentes sitios. Esto me permite tener los datos disponibles en distintos sitios y realizar queries en paralelo.

Las desventaja de esto es que será necesario un constante actualizado y sincronizado de datos para mantener una copia exacta de la base de datos. Esto genera un complicado control de concurrencias.

Podemos tener **replicación completa** donde toda la base de datos se encuentra en todos los sitios o podemos tener **replicación parcial** en donde algunos fragmentos de datos son replicados y otros no

Estrategias de replicación para datos

	Descripción	Utilidad	Desventaja
Replicación transaccional	Los usuarios reciben una copia inicial completa y van recibiendo actualizaciones a medida que los datos cambian. Funciona con un sistema publicador-suscriptor para garantizar la consistencia.	Se utiliza en entornos servidor-servidor.	Todos deben tener el mismo schema de datos. Constantemente se están enviando datos de un lado a otro.
Replicación Snapshot	Distribuye una copia exacta de la base de datos en un momento dado y no chequea por cambios en los datos.	Generalmente se usan cuando los cambios en los datos no son frecuentes.	Más lento que el anterior dado que tiene que enviar instantáneas de un punto a otro.
Replicación merge	Los datos de 1 o más bases de datos se combinan en una sola base de datos.	Es el tipo de replicación más complejo porque permite al publicador y al suscriptor realizar cambios en la base de manera independiente.	Se usa en entornos clientes-servidor

Fragmentación

Tablas fragmentadas significa que las tablas están almacenadas en pequeños fragmentos y cada fragmento se almacena en sitios diferentes según sea requerido.

El requisito es que los fragmentos luego puedan ser reconstruidos en una tabla original sin perder datos.

Fragmentación horizontal

Se divide una tabla horizontalmente asignando a cada fila o a un grupo de fila uno o más fragmentos. Estos fragmentos se asignan a distintos lados de una base de datos distribuida

Eno	Ename	Design	Salary	Dep
101	A	abc	3000	1
102	B	abc	4000	1
103	C	abc	5500	2
104	D	abc	5000	2
105	E	abc	2000	2

Fragmentación horizontal

T1				
Eno	Ename	Design	Salary	Dep
101	A	abc	3000	1
102	B	abc	4000	1

T2				
Eno	Ename	Design	Salary	Dep
103	C	abc	5500	2
104	D	abc	5000	2
105	E	abc	2000	2

Fragmentación vertical

Es el proceso de descomponer las columnas. Esta fragmentación es útil cuando algunos sitios no necesitan todas las columnas. Para recomponer la tabla, es necesario que se tenga la primary key. Dado que este proceso debe recomponerse con un JOIN natura, es necesario agregar una tupla-id para lograrlo

Eno	Ename	Design	Salary	Dep
101	A	abc	3000	1
102	B	abc	4000	1
103	C	abc	5500	2
104	D	abc	5000	2
105	E	abc	2000	2

Fragmentación vertical

Eno	Ename	Design	Tuple_id
101	A	abc	1
102	B	abc	2
103	C	abc	3
104	D	abc	4
105	E	abc	5

Fragmentación vertical

Salary	Dep	Tuple_id
3000	1	1
4000	1	2
5500	2	3
5000	2	4
2000	2	5

Fragmentación Mixta

Se realiza la fragmentación en ambos sentidos.

- Se realiza primero una fragmentación horizontal y luego una fragmentación vertical
- Se realiza una fragmentación vertical y luego una fragmentación horizontal

Ename	Design
A	abc
B	abc
C	abc

Ventajas y desventajas generales

- Ventajas
  - Incrementa la resiliencia y disminuye los riesgos dado que trabaja con réplicas de los mismos datos en múltiples instancias por lo que si se cae un nodo, la aplicación puede seguir trabajando.
  - Distribuir la base de datos puede mejorar la performance ya que todo el trabajo en varias bases de datos evita un cuello de botella por tener que realizar todo en la misma pc.
  - Distribuir la base de datos geográficamente permite reducir la latencia

- Desventajas
  - Incrementa la complejidad operacional
  - Incremento en la curva de aprendizaje

Ventajas y desventajas de replicación

- Ventajas
  - Mejora del performance ya que los datos pueden leerse de una copia local en vez de una remota.
  - Incrementa la disponibilidad de datos ya que permite tenerlos en caso de que falle la base principal.
  - Mejora la escalabilidad ya que la carga en la base primaria se reduce mediante la lectura de copias.

- Desventajas
  - Incrementa la complejidad ya que el proceso de replicación necesita configurarse y mantenerse
  - Incrementa el riesgo de inconsistencias ya que los datos pueden ser actualizados en distintas réplicas
  - Incrementa el consumo de almacenamiento y uso de red ya que las copias deben ser almacenadas y transmitidas

Ventajas y desventajas de fragmentación

- Ventajas
  - Aumenta la eficiencia si los datos están cerca del sitio de uso
  - Optimización de queries locales pueden ser suficiente para algunas queries ya que los datos están disponibles localmente
  - Facilita mantener la seguridad y la privacidad ya que los datos se encuentran fragmentados

- Desventajas
  - Velocidad de acceso puede ser muy alta si se necesitan datos de distintos fragmentos

Comparación

No hay mucho para comparar dado que se está hablando de un tipo de arquitectura de base de datos

Cuando usar

- Cuando queremos tener una buena disponibilidad (probabilidad de correr continuamente durante un intervalo de tiempo) y fiabilidad (probabilidad de correr en determinado tiempo)

- Cuando los tiempos de respuesta son importantes. Dado que podemos fragmentar los datos y los usuarios pueden realizar solicitudes donde los únicos datos que se necesiten sean los locales

Bases de datos distribuidas gratis

Bases de datos pagas - Amazon SimpleDB

Free Tier*

You can get started with Amazon SimpleDB for free. New and existing customers receive 25 SimpleDB Machine Hours and 1 GB of Storage for free each month. Many applications should be able to operate perpetually within these free tier limits.

Machine Utilization

Region: South America (Sao Paulo)

- $0.00 for first 25 hours
- $0.19 per machine hour over 25 hours

Amazon SimpleDB measures the machine utilization of each request and charges based on the amount of machine capacity used to complete the particular request (SELECT, GET, PUT, etc.), normalized to the hourly capacity of a circa 2007 1.7 GHz Xeon processor. See below for a more detailed description of how machine utilization charges are calculated.

Data Transfer**

Region: South America (Sao Paulo)

Data Transferred	Pricing
Data Transfer IN	
All data transfer in	$0.00 per GB
Data Transfer OUT ***	
Next 9.999 TB / Month	$0.15 per GB
Next 40 TB / Month	$0.138 per GB
Next 100 TB / Month	$0.126 per GB
Greater than 150 TB / Month	$0.114 per GB

Data transfer "in" and "out" refers to transfer into and out of Amazon SimpleDB. There is no additional charge for data transferred between Amazon SimpleDB and other Amazon Web Services within the same Region (i.e., $0.00 per GB). Data transferred across Regions (e.g., between Amazon SimpleDB in the EU (Ireland) Region and Amazon EC2 in the US East (Northern Virginia) Region, will be charged at Internet Data Transfer rates on both sides of the transfer.

Structured Data Storage

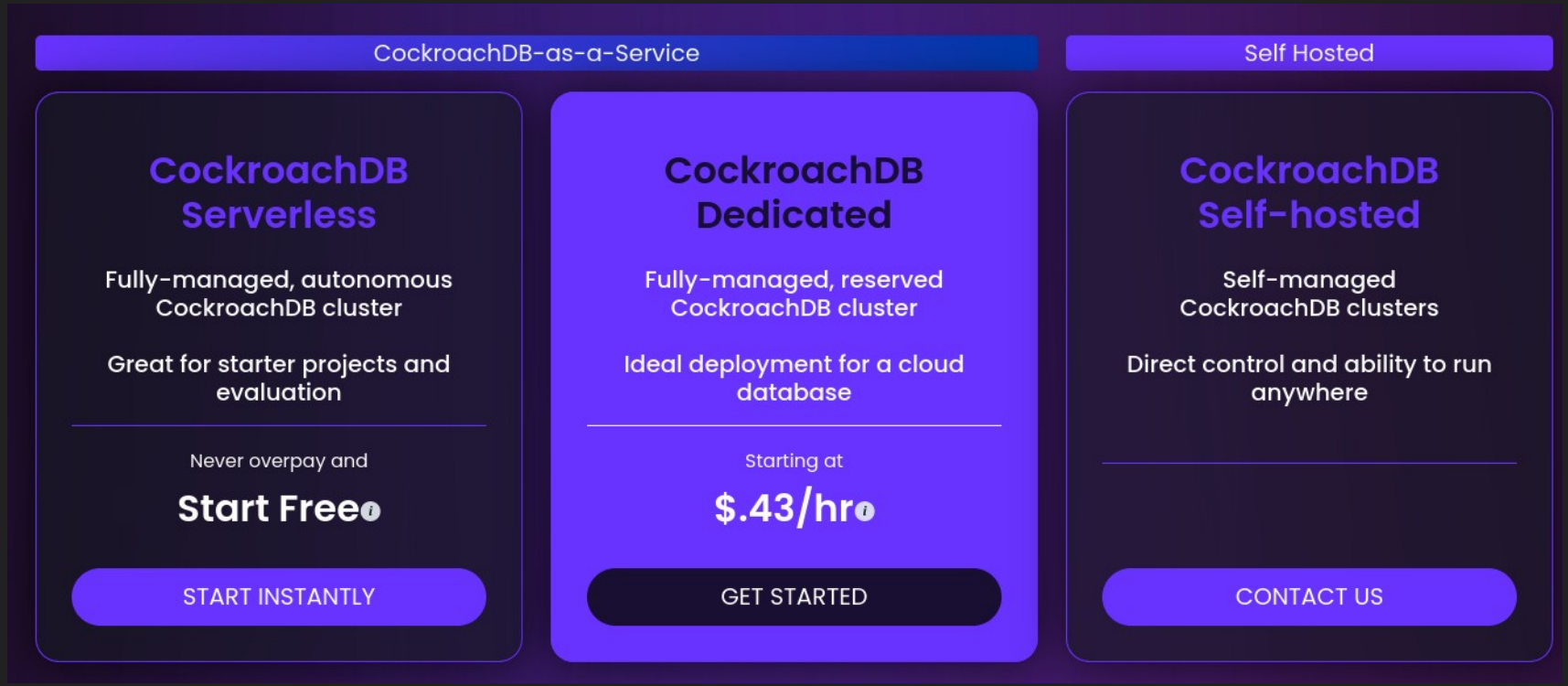
Region: South America (Sao Paulo)

- $0.00 for first GB-month
- $0.34 per GB-month thereafter

Amazon SimpleDB measures the size of your billable data by adding the raw byte size of the data you upload + 45 bytes of overhead for each item, attribute name and attribute-value pair.

Amazon SimpleDB is designed to store relatively small amounts of data and is optimized for fast data access and flexibility in how that data is expressed. In order to minimize your costs across AWS services, large objects or files should be stored in Amazon S3, while the pointers and the meta-data associated with those files can be stored in Amazon SimpleDB. This will allow you to quickly search for and access your files, while minimizing overall storage costs. Click here for a detailed explanation of how storage in Amazon SimpleDB and storage in Amazon S3 differ and a more detailed description on calculating your Storage Costs.

Base de datos pagas - CockroachDB



Links de instalación

Instalación de Cassandra

https://linuxhint.com/install-use-apache-cassandra-ubuntu-22-04/

Configuración para funcionar como base de datos distribuida

https://docs.datastax.com/en/cassandra-oss/3.0/cassandra/initialize/initSingleDS.html

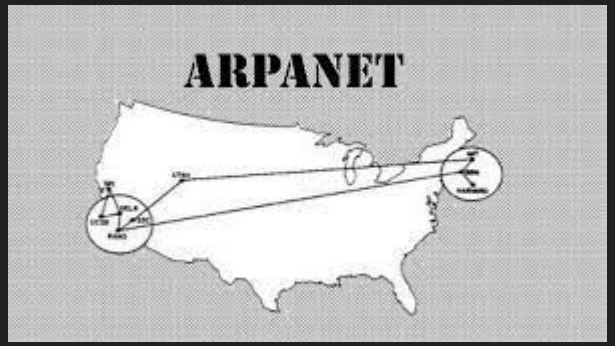
Historia

Durante los años 60, en plena guerra fría, un investigador llamado Paul Baran desarrolló una idea de comunicación distribuida en donde los mensajes viajan a través de una red de nodos hasta que finalmente llegan a su destino.

Estos nodos serán computadoras en vez de switches telefónicos que crearian múltiples caminos y fragmentarian el mensaje de tal manera que su transmisión fuera sencilla y eficiente.

Historia

Con el objetivo de lograr la mayor eficiencia posible en las instalaciones, se creó un proyecto llamado ARPANET que tenía como objetivo conectar todas las instalaciones a una sola red.



A medida que la red fue creciendo, se empezaron a investigar diferentes tipos de redes. En conjunto con Europa, se empezaron a crear protocolos y sistemas de direccionamiento que permitirían la aparición de Internet

Historia

Los primeros programas distribuidos se conocieron como Creeper y Reaper.

Creeper utilizaba los ciclos de procesador libres para copiarse al próximo sistema y desaparecer del anterior. Más adelante fue modificado para que solo se copiará.

Por este motivo nació Reaper que tenía como objetivo borrar todas las copias de Creeper.

En 1988, el DEC System Research Center comenzó un proyecto donde le enviaba a voluntarios, una aplicación que debían ejecutar en momentos libres y luego reenviar esos resultados. Este proyecto llegó a tener cerca de 100 usuarios en 1990