Sistema de archivos de Google

Mario Alonso Carmona Dinarte
A71437

Agenda

- Introducción

- Definición GFS

- Supuestos

- Diseño & Caracteristícas

- Ejemplo funcionamiento (paso a paso)

- Caracteristicas del Hadrware

- Conclusiones

Introducción

Definición

Supuestos

- El sistema esta construido de muchos componentes básicos baratos que siempre fallan. Este debe constantemente monitorearse y detectar, tolerar y recuperase prontamente de fallas de componenes como parte de su rutina.

- El sistema almacena un modesto número de archivos grandes. Se esperan unos cuantos millones de archivos, cada uno tipicamente de 100 MB o más grandes. Archivos Multi-GB son el caso común and deben ser manejados eficientemente. Archivos pequeños deben ser soportados, pero no hay necesidad en su optimización.

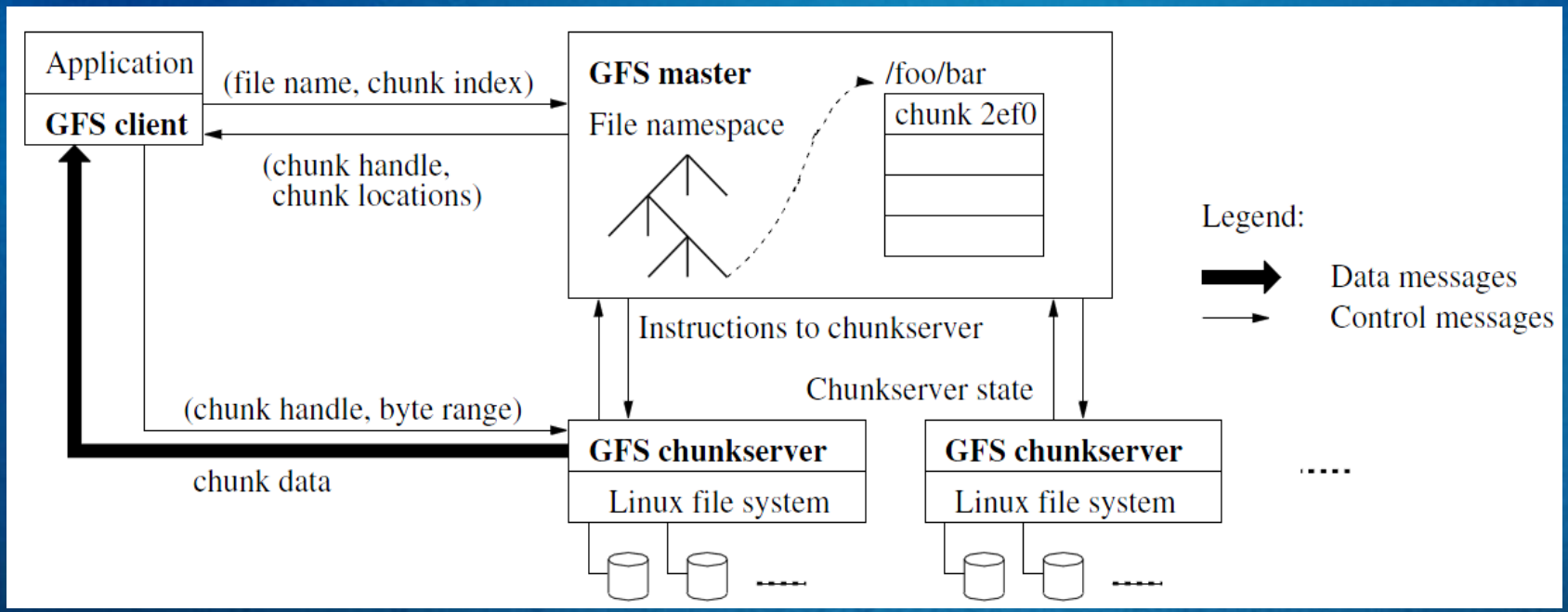
- Las cargas de trabajo consisten en dos tipos de lecturas:
  - Largas lecturas de tranmisión
  - Pequeñas lecturas aleatorias

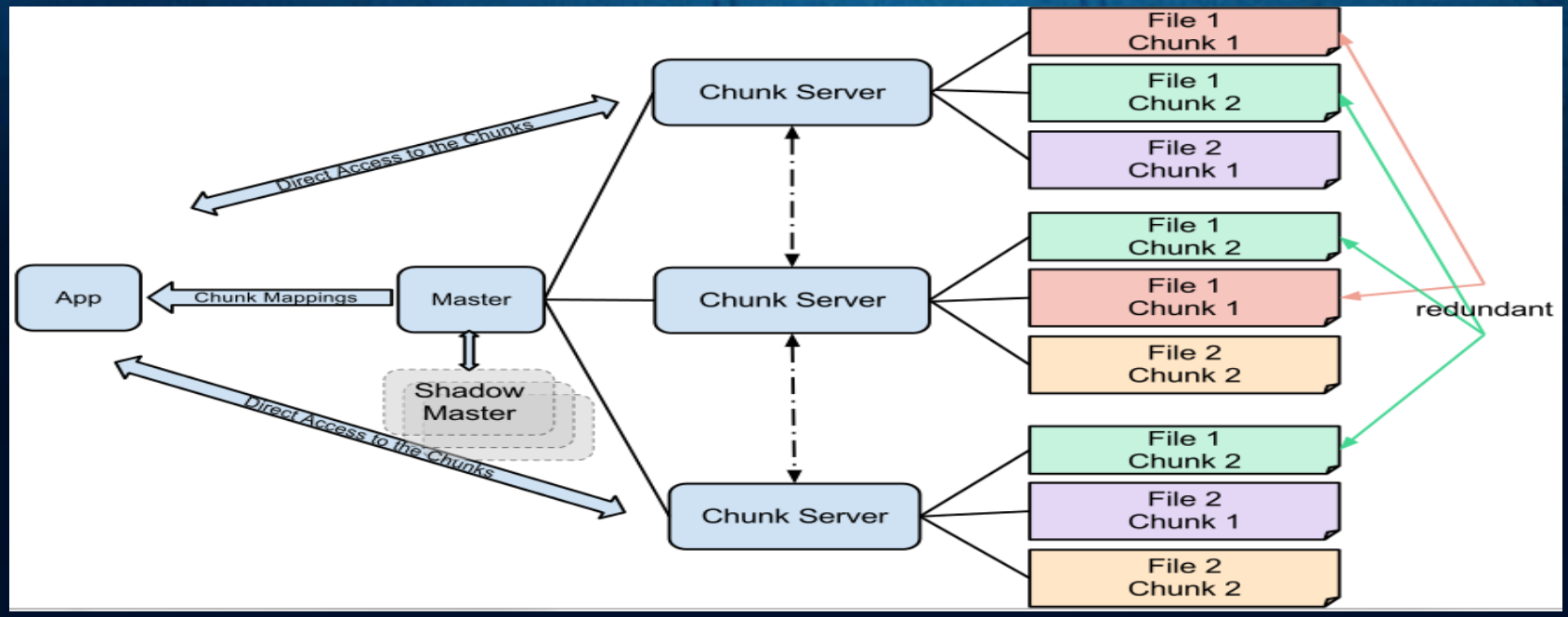
Supuestos

- Las cargas de trabajo también tiene muchas escrituras grandes y secuenciales que anexan datos a los archivos.
- El sistema debe eficientemente implementar una semántica bien definida para muchos clientes que anexan concurrentemente al mismo archivo. Atomicidad con carga adicional de los recursos (overhead) minima por sincornización es esencial.
- Ancho de banda alto y prolognado es más importante que baja latencia. La mayoría de las aplicaciones ponen primero el procesamiento de datos volumen a una alta velocidad, que un rápida respuesta.

Diseño & Caracteristícas

Arquitectura





- Linux
-
- 64MB chunks
- 3 Replicas
- 64 bit chunkhandler
- HeartBeat Messages
- No cache file data

Operaciones

- Create
- Open
- Read
- Write

- Append:
  - Los clientes especifican solo los datos
  - GFS elige el offset

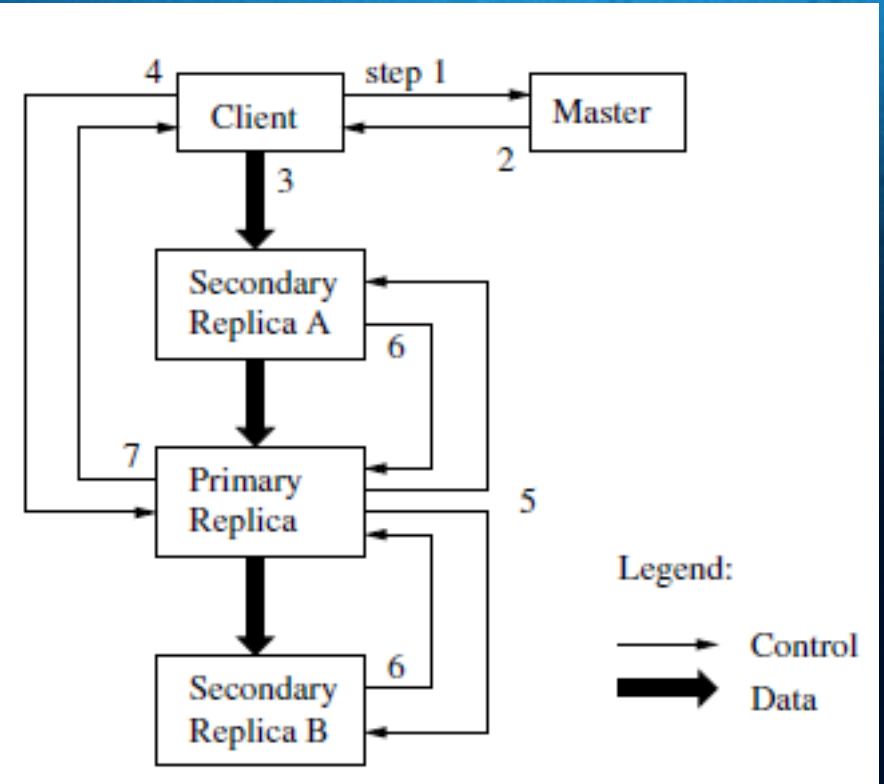
- Snapshot
  - Crear copias de grandes conjuntos de datos
  - Crear checkpoints
  - Mismo disco (+ rápido, - ancho de banda)

Funcionalidades

- Bitácora de operaciones (Operation log)

- "Leases" y orden de las mutaciones

- Flujo de datos (pipeline)
  - B/T + RL: 1MB/100Mbps + 2*1ms= 80ms



Funcionalidades

- Recolección de Basura (cuando un archivo es eliminado)

- Detección de replicas obsoletas

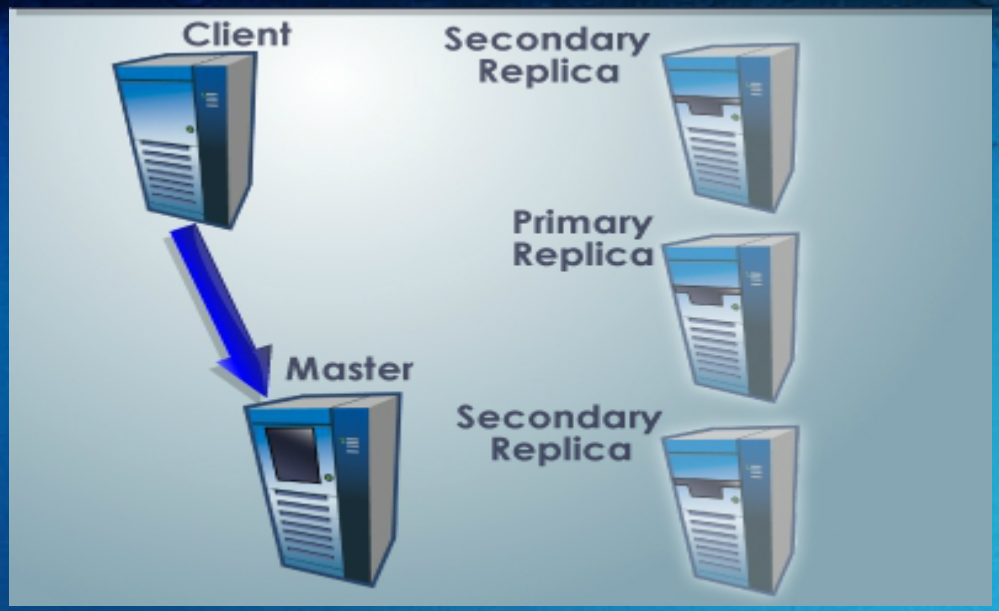
- Creation, Re-replicación, Rebalanceo

- Integridad de los datos
  - Shadow Master
  - Chunks: Bloques de 64KB - 32bit checksum
  - Verificación en periodos de inactividad

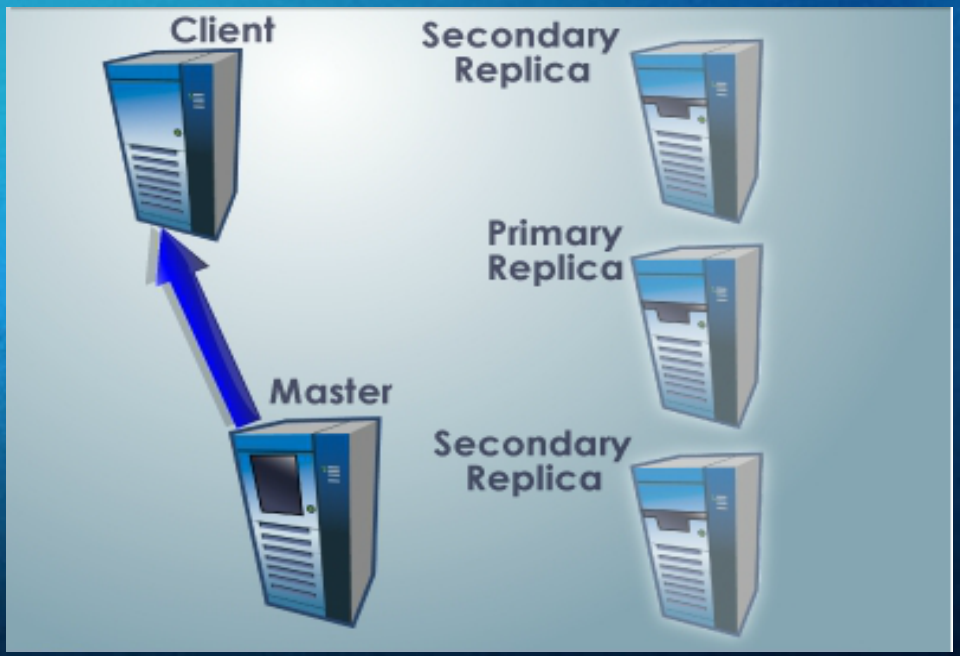
- Fast Recovery(Segundos - Terminación)

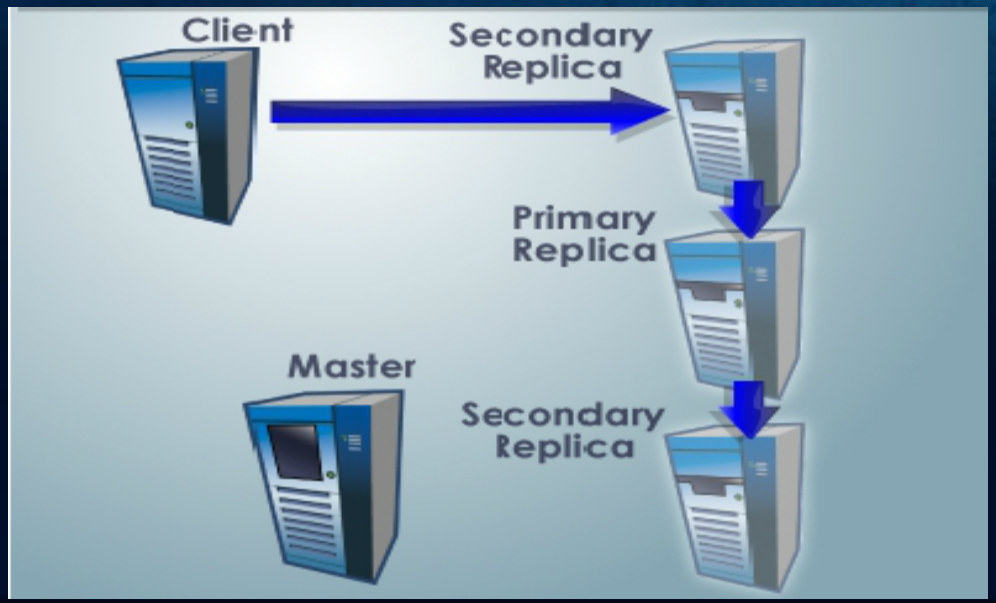
Ejemplo funcionamiento

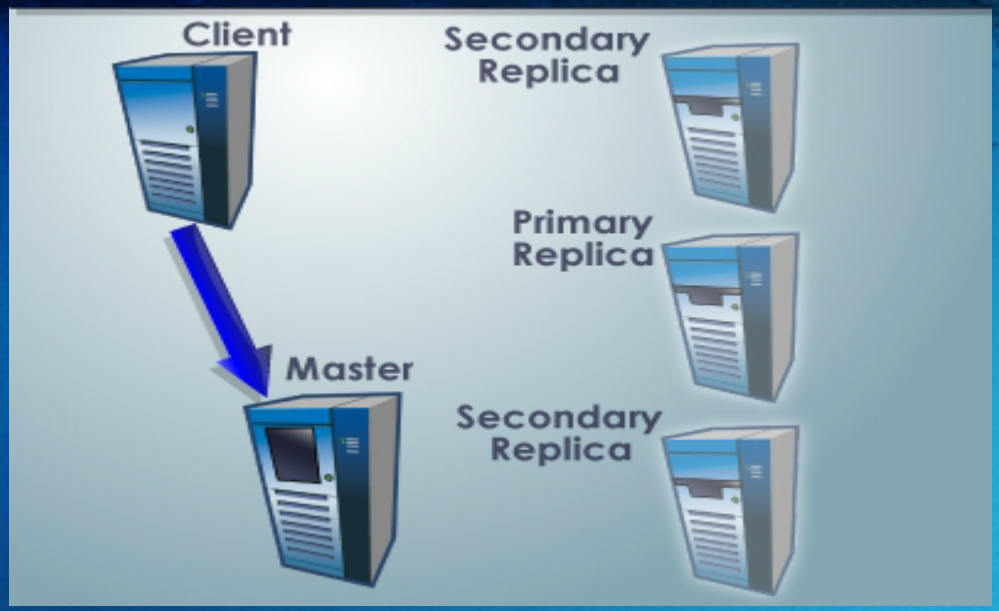
Explicación paso a paso escritura



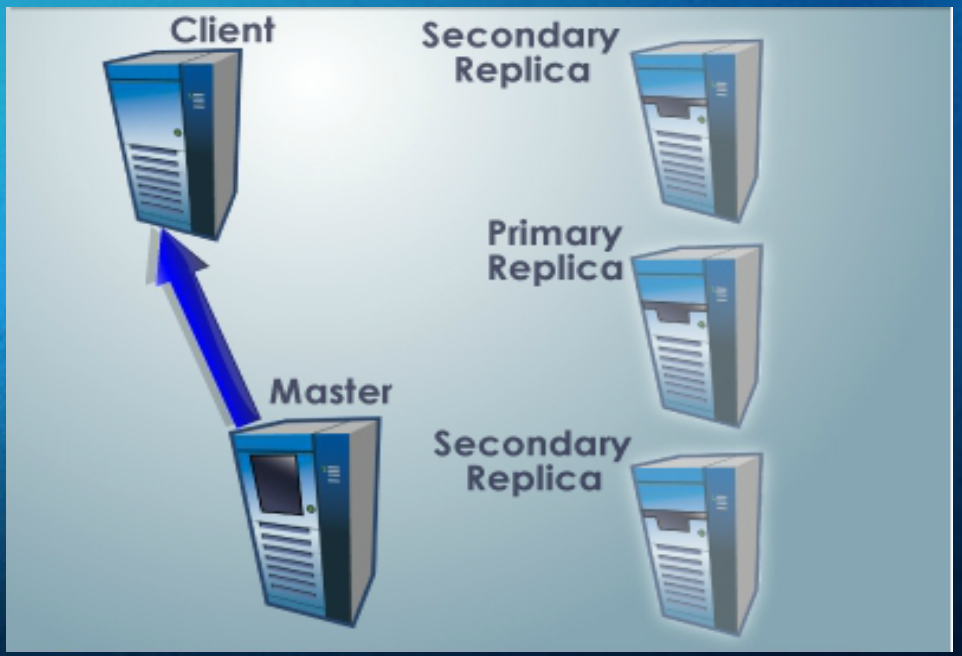
El cliente envía una solicitud al Master para encontrar la localización del servidor de trozos que actua como la replica primaria.

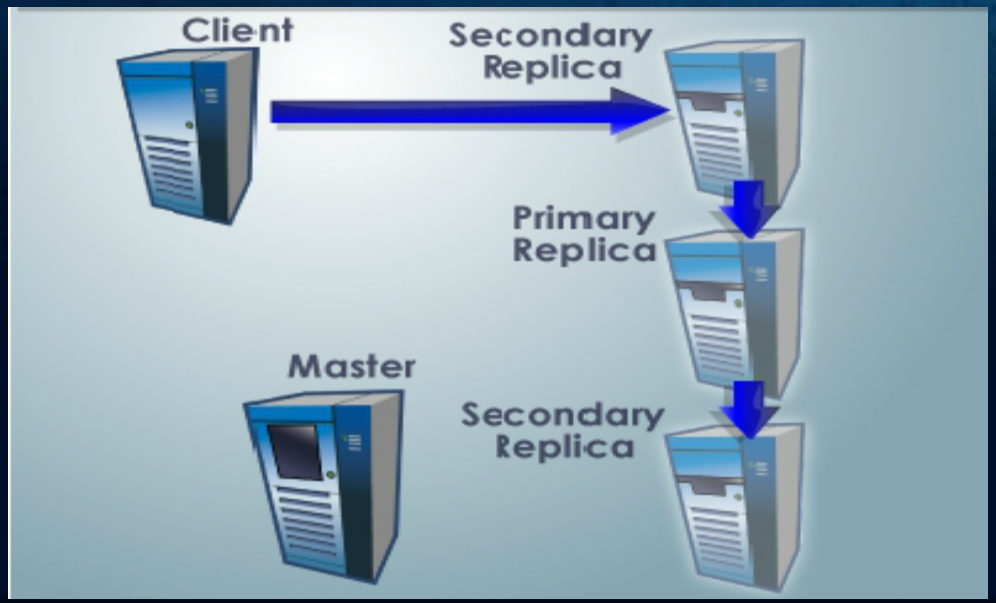


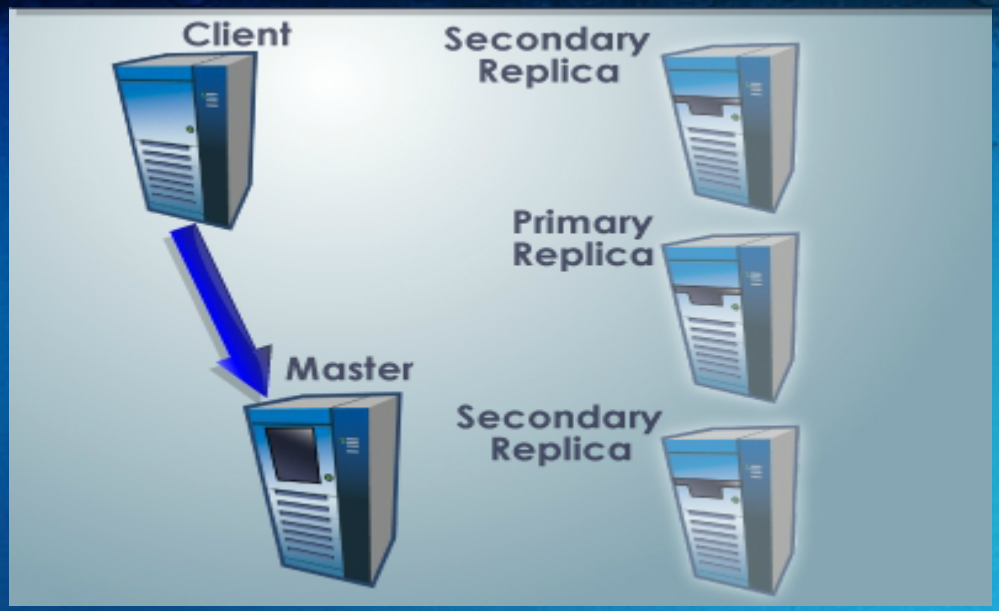


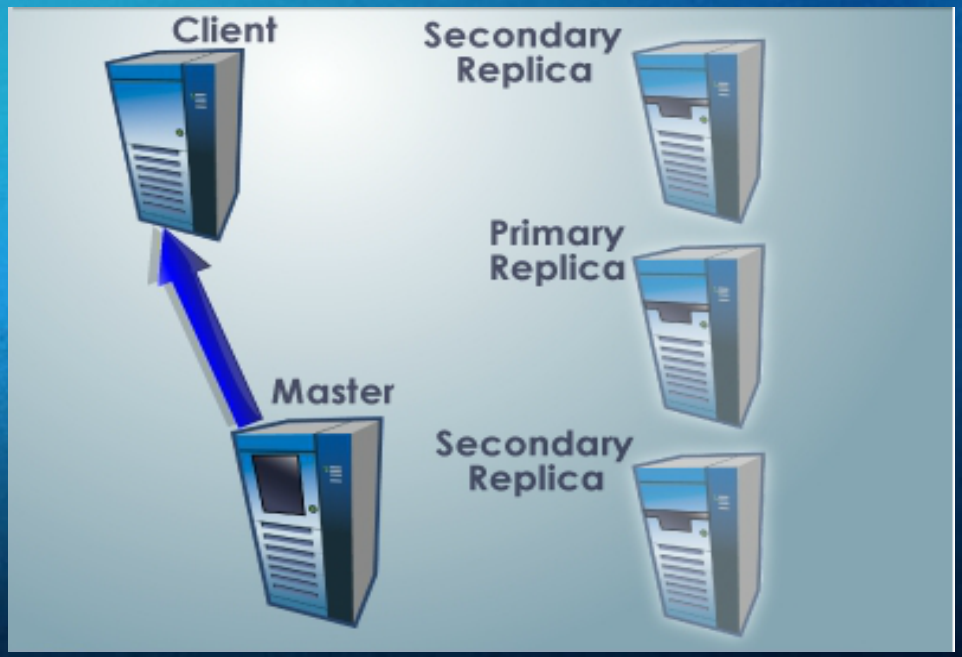


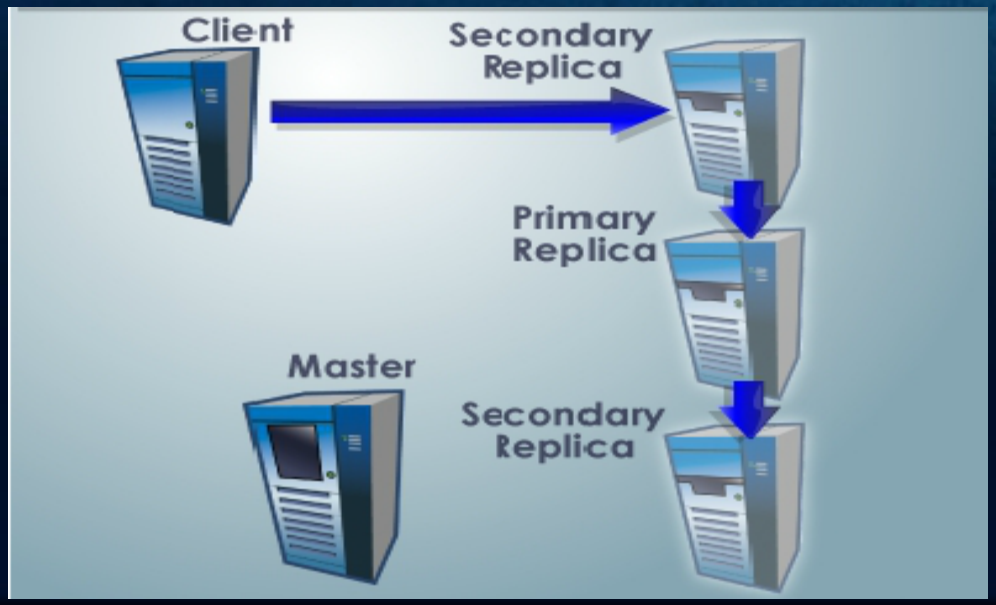
El Master envía al cliente la localización de las replicas de servidores de trozos e identifica la replica primaria.



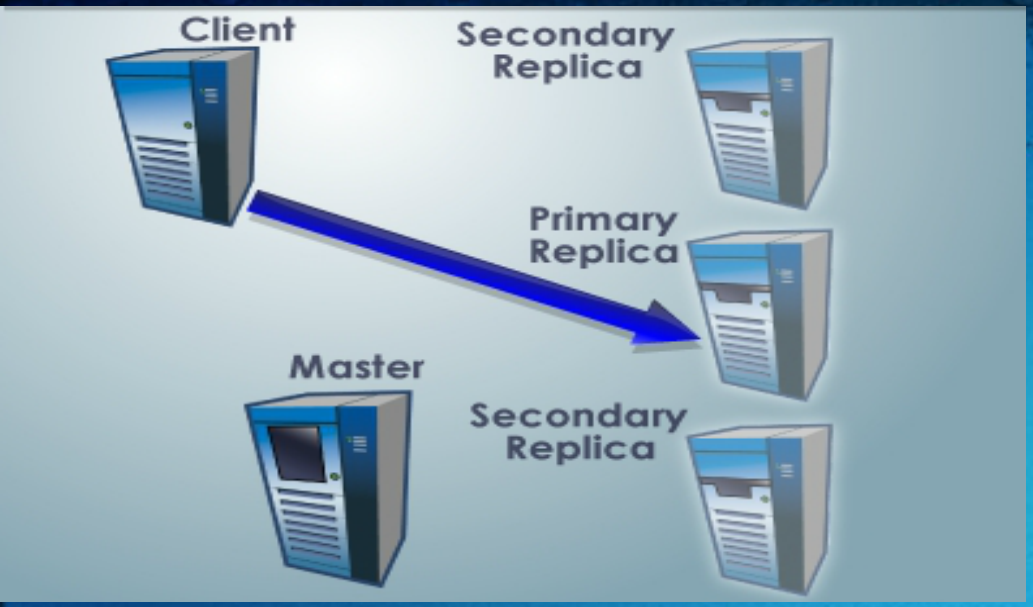




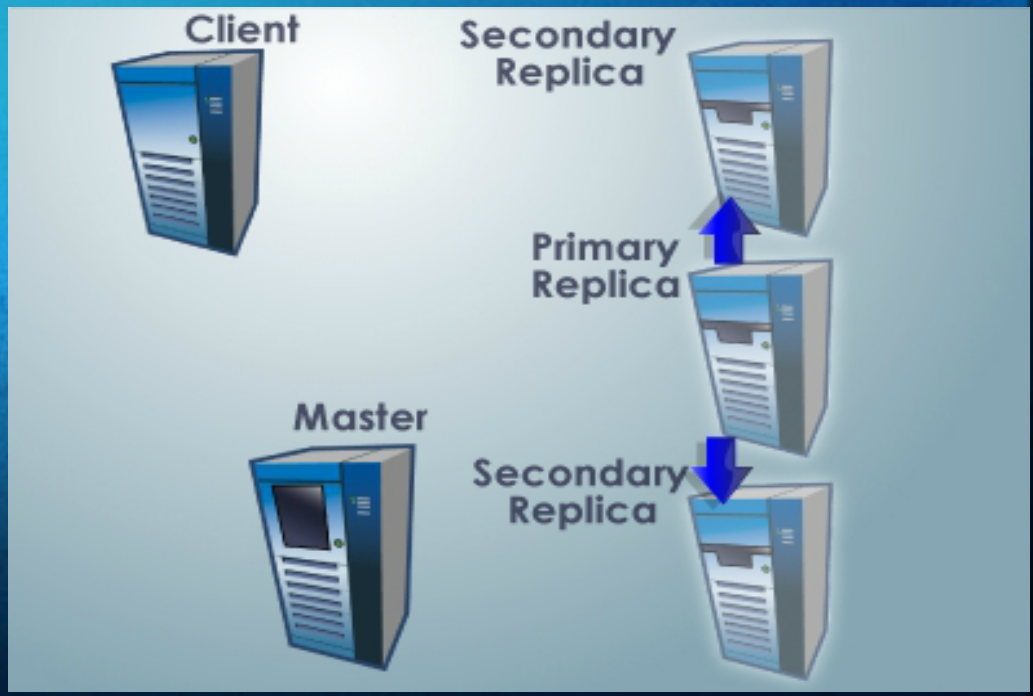


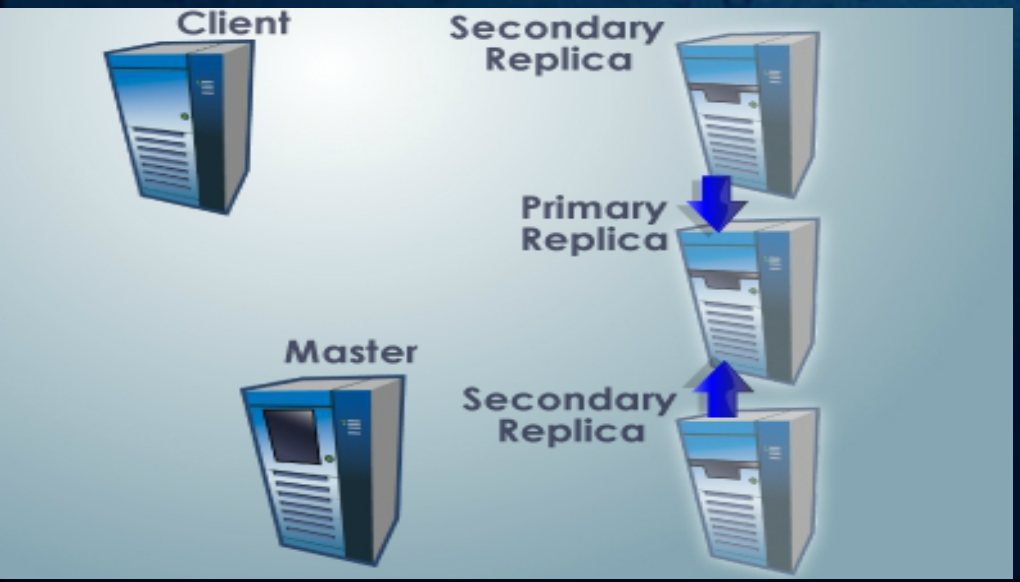


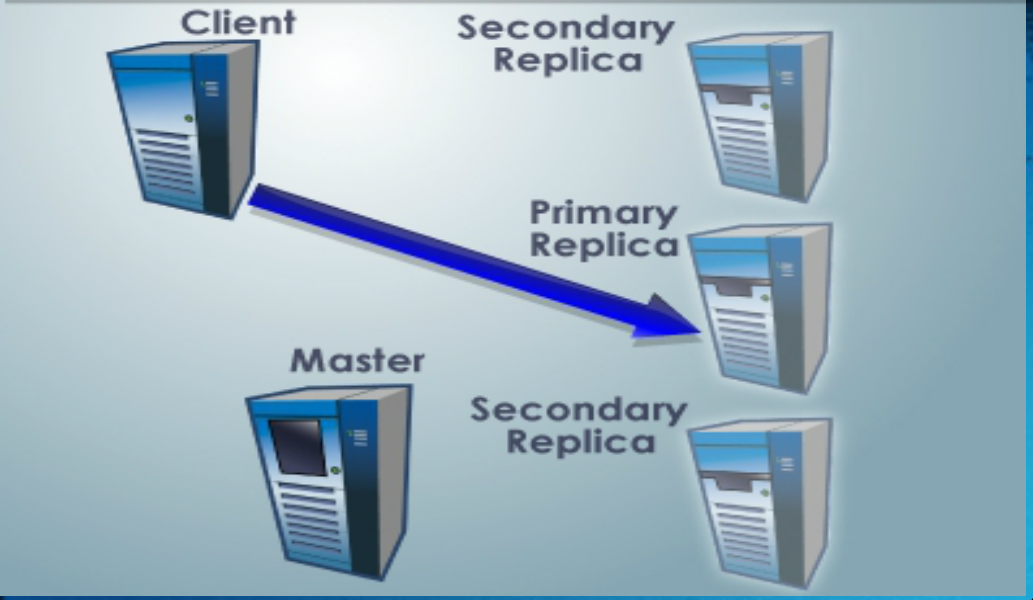
El cliente envía los datos a escribir a todos los servidores de trozos, iniciando con el más cercano y aplicando "pipeline" de datos en el flujo.



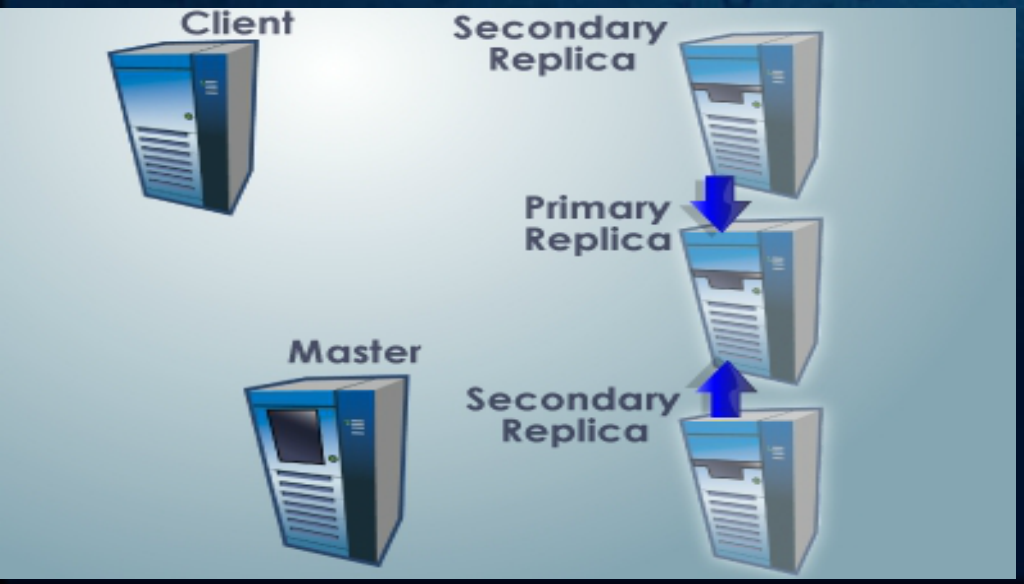
Una vez que las replicas reciben los datos, el cliente le dice a la replica primaria que inicie con con función de escritura.

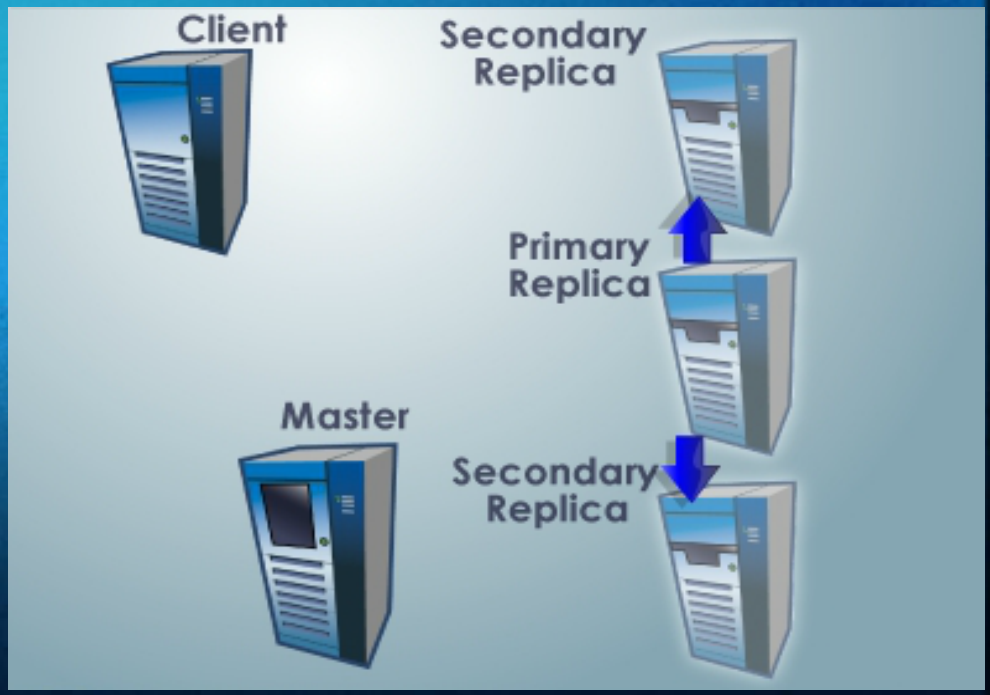


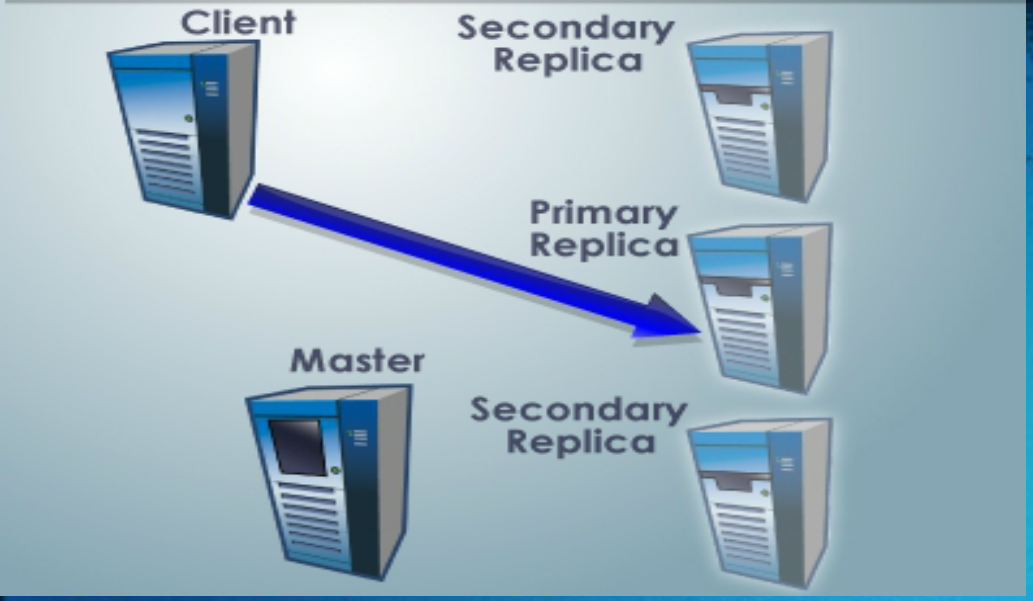


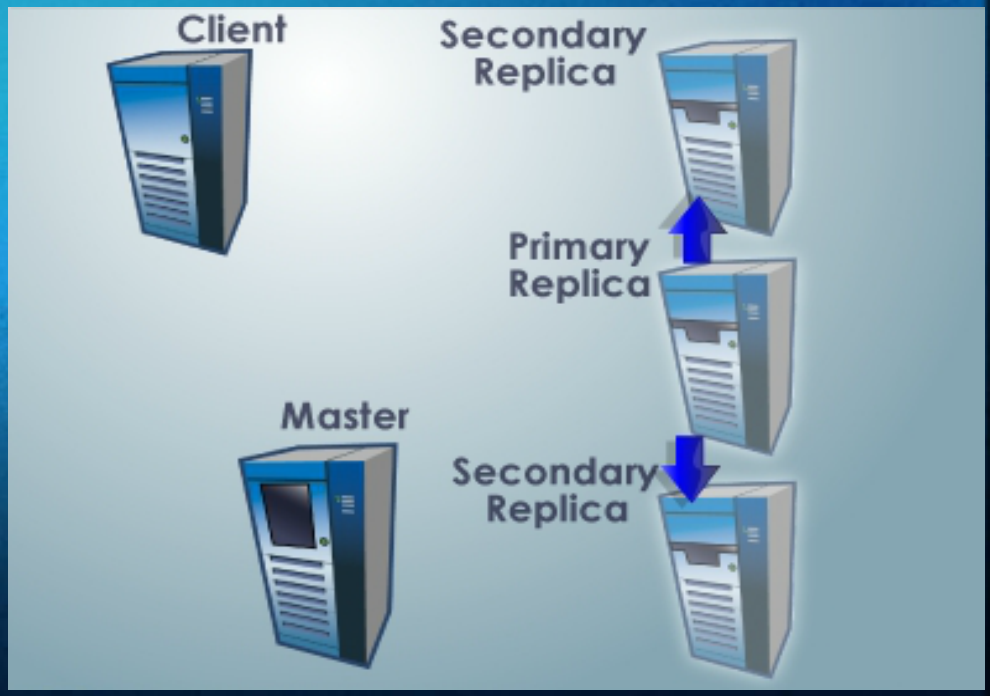


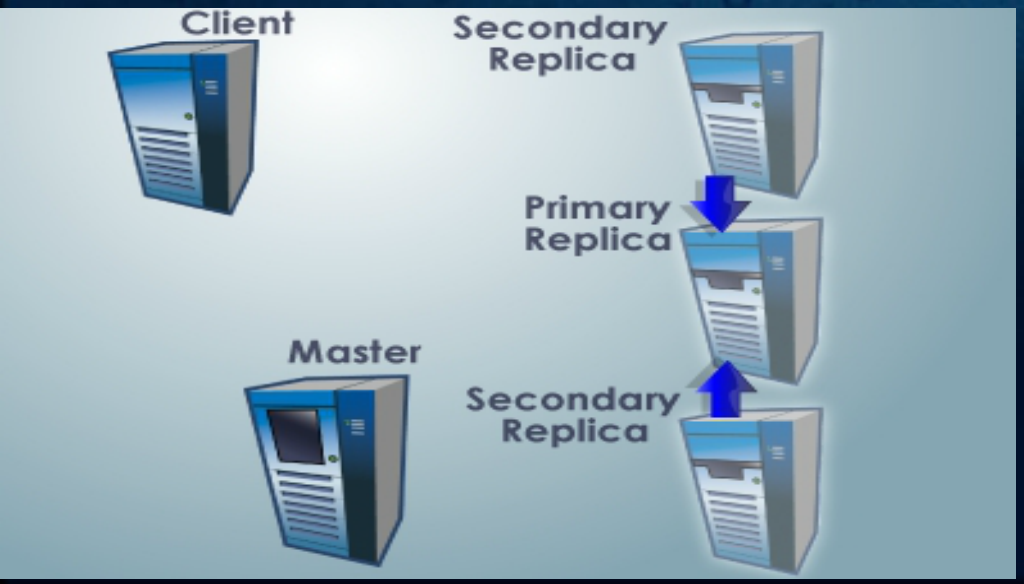
La replica primaria escribe los datos en el trozo adecuado, luego le dice a las otras replica que hagan lo mismo.



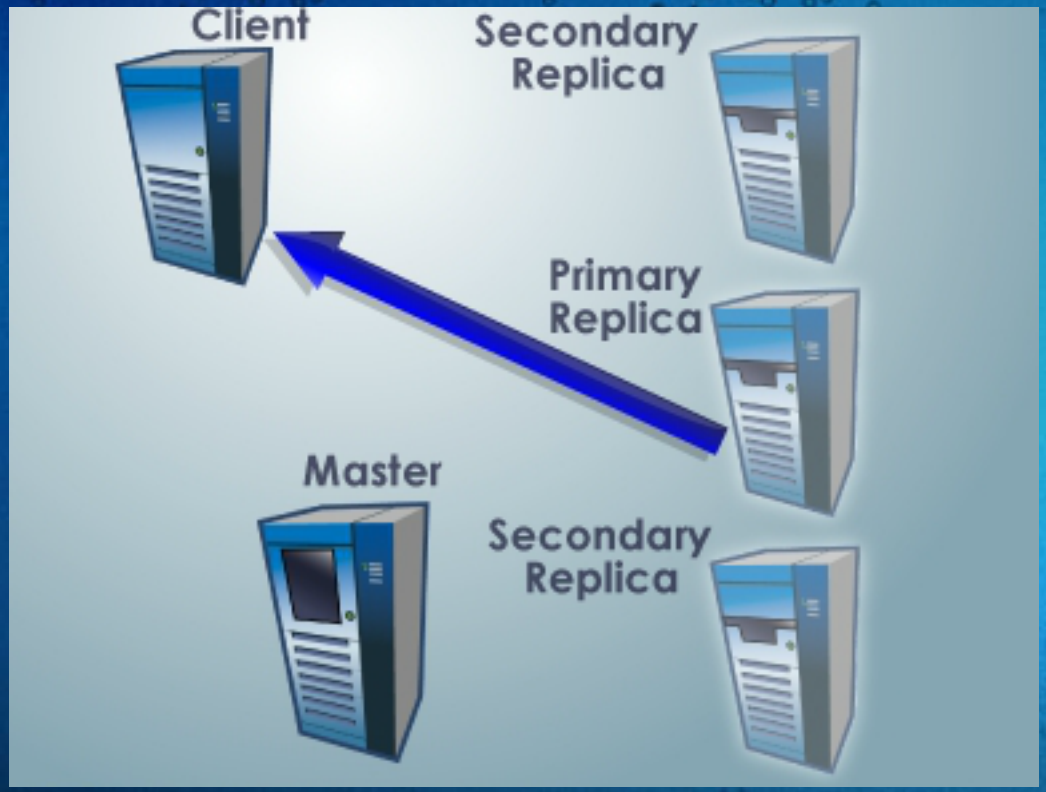








Las replicas secundarias completan la función de escritura y se lo reportan a la replica primaria.



La replica primaria envía la confirmación a el cliente.

Fin del proceso

Caracteristicas de Hardware

- Dual 1.4 gigahertz Pentium III

- 2GB Ram memory

- 80GB disco duro

- Maquinas, Conexion Ethernet Full Duplex de 100 Mbps

- Switches HP 2524, conectados con una conexion de 1Gbps

Conclusiones

- No se necesita un hardware de última generación para tener un sistema eficiente
- Se necesita un monitoreo constante en este tipo de sistemas
- La replicación es un precio justo a pagar por disponibilidad
- Transmisión directa de datos mejora la eficiencia y evita cuellos de botella
- Manejo centralizado de meta-datos da grandes ventajas al sistema en diversos aspectos

Bibliografía

[1] Ghemawat S, Gobioff H, Lueng S. (2003). “The Google File System”. 19th ACM Symposium on Operating Systems Principles, Lake George, NY, October, 2003

[2] Strickland J. “How Google File System Works”.Obtenido el 14 de Enero de 2012 de : http://computer.howstuffworks.com/internet/basics/google-file-system5.htm

[3] Sin Autor. (2012). “How Google File System Works”. Obtenido el 14 de Enero de 2012 de : http://en.wikipedia.org/wiki/Google_File_System

Preguntas - Comentarios

Muchas Gracias!