GFS:

**Modificações concorrentes de arquivos:**

As mutações de dados podem ser gravações ou acréscimos de registro. Uma gravação faz com que os dados sejam gravados em um deslocamento de arquivo especificado pelo aplicativo. Um acréscimo de registro faz com que os dados (o “registro”) sejam anexados atomicamente pelo menos uma vez, mesmo na presença de mutações simultâneas, mas em um deslocamento de escolha do GFS. (Em contraste, um acréscimo “regular” é meramente uma gravação em um deslocamento que o cliente acredita ser o final atual do arquivo.)

O deslocamento é retornado ao cliente e marca o início de uma região definida que contém o registro. Além disso, o GFS pode inserir preenchimento ou registrar duplicatas no meio. Eles ocupam regiões consideradas inconsistentes e normalmente são ofuscados pela quantidade de dados do usuário.

**Replicação de arquivos:**

Por padrão, armazenamos três réplicas, embora os usuários possam designar diferentes níveis de replicação para diferentes regiões do namespace do arquivo. O mestre mantém todos os metadados do sistema de arquivos.

**Heterogeneidade:**

A Google utiliza o GFS para organizar e manipular grandes arquivos e permitir que aplicações consigam usar os recursos necessários. Exemplo de aplicações que usam estes dados: YouTube, Google Earth, Blogger, GMail, Orkut, Google Maps, Google Sugest, Google Desktop Search, entre outras.

**Tolerância a Falhas:**

O seu processo para tolerância especializa-se na recuperação e replicação, caracterizados por:

• Rápido retorno: permite que os servidores de porção e o mestre restabeleçam seus processos em segundos, não importando como eles tenham parado;

• Replicação de porção: é responsável em replicar as porções em diferentes níveis para diferentes partes do espaço de nomes;

• Replicação do mestre: faz réplicas do estado do servidor mestre em vários computadores, permitindo que um novo servidor possa ser estabelecido em caso de falhas com o anterior.

**Consistência:**

Para garantir a consistência, o cliente recebe a localização (offset) do seu registro concatenado no arquivo após a conclusão da escrita, ao invés de escolhe-lo durante a operação. Escritas comuns em regiões arbitrarias são permitidas, porém não garantem a consistência no caso em que ocorrer uma concorrência.

Basicamente, o mestre pega uma réplica como primária e permite as mutações nela. Esta cópia define a ordem das mutações que após são seguidas pelas réplicas secundárias.

**Proteção:**

Outros metadados incluem propriedade e permissões de arquivos, mapeamento de arquivos para partes e a versão atual de cada parte. Além disso, para cada fragmento, armazenamos os locais de réplica atuais e uma contagem de referência para implementar a cópia na gravação.

A criação de arquivo não requer um bloqueio de gravação no diretório pai porque não há um “diretório” a ser protegido contra modificação. O bloqueio de leitura do nome é suficiente para proteger o diretório pai da exclusão.

**Eficiência:**

De acordo com os resultados do benchmarking, quando usado com um número relativamente pequeno de servidores, o sistema de arquivos atinge um desempenho de leitura comparável ao de um único disco (80–100 MB/s), mas tem um desempenho de gravação reduzido (30 MB/s) e é relativamente lento (5 MB/s) em anexar dados a arquivos existentes. Os autores não apresentam resultados sobre o tempo de busca aleatória.

Como o nó mestre não está diretamente envolvido na leitura de dados (os dados são passados do servidor chunk diretamente para o cliente de leitura), a taxa de leitura aumenta significativamente com o número de servidores chunk, atingindo 583 MB/s para 342 nós. A agregação de vários servidores também permite grande capacidade, embora seja um pouco reduzida pelo armazenamento de dados em três locais independentes (para fornecer redundância).

**Transparência (GFS)**

**De acesso**

O acesso é feito a partir de API’s relativas ao GFS fornecidas pelo próprio google. As conexões TCP são usadas para comunicação. A tubulação é usada para transferência de dados através de conexões TCP.

**De localização**

Os arquivos são organizados hierarquicamente em diretórios e identificados por “path names”

**De mobilidade**

**De desempenho**

**De escalabilidade**

**Contra falhas**

**De replicação**